DOCUMENT RESUME

ED 085 590 95 CE 000 787

AUTHOR Goldman, Charles I.
TITLE Local Education Agency Planning Analyst's Procedures. A Vocational Education Planning System for Local School Districts. Volume III.
INSTITUTION Government Studies & Systems, Philadelphia, Pa.
SPONS AGENCY New Jersey State Dept. of Education, Trenton. Div. of Vocational Education.
PUB DATE Jun 73
NOTE 104p.; For related documents, see CE 000 785-6, 788 and CE 000 731-5

EDRS PRICE MF-$0.65 HC-$6.58
DESCRIPTORS Annual Reports; Computer Oriented Programs; *Computer Programs; Cost Effectiveness; *Data Analysis; Data Bases; *Data Processing; Educational Planning; Planning; Prediction; Program Costs; Programing Languages; *Program Planning; Vocational Education

ABSTRACT

The manual is part of a series to assist in planning procedures for local and State vocational agencies. It details steps required to process a local education agency's data after the data have been coded onto keypunch forms. Program, course, and overhead data are input into a computer data base and error checks are performed. A computer model is run using the data base to produce current year reports and five year projections of manpower needs met and student demand served, facilities needed versus those available, teachers required and available, direct program costs (salaries, materials, travel), equipment and maintenance, total overhead costs by account. The procedures are written using Rapid Access Management Information System (RAMIS). The manual assumes data will be input and the model run using the National CSS, Inc., computer service. (MS)

A Vocational Education Planning System
FOR LOCAL SCHOOL DISTRICTS

NEW JERSEY STATE DEPARTMENT OF EDUCATION
DIVISION OF VOCATIONAL EDUCATION
225 WEST STATE STREET
TRENTON, N.J. 08625

with the assistance of
GOVERNMENT STUDIES & SYSTEMS, INC.
3401 MARKET STREET
PHILA., PA. 19104

ED 085590

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

LOCAL EDUCATION AGENCY PLANNING ANALYSTS' PROCEDURES

Vol. III

CE 000 787

FILMED FROM BEST AVAILABLE COPY

ED 085590

A VOCATIONAL EDUCATION PLANNING SYSTEM FOR LOCAL SCHOOL DISTRICTS

Volume III: Local Education Agency Planning Analyst's Procedures

Produced For

Edison Township

Linden

Lower Camden County Regional High School District

Middlesex County Vocational Schools

Somerset County Vocational School and Technical Institute

and

The State Department of Education
Division of Vocational Education

With the Assistance of

Government Studies and Systems, Inc.

The project presented herein was performed pursuant to a grant from the New Jersey State Department of Education, Division of Vocational Education under Public Law 90-576, Part C, Section 131, (b).

July 1970 - June 1973

Acknowledgments

The Division of Vocational Education of the New Jersey State Department of Education has long recognized the need to introduce more science into the art of educational planning. This publication is an outgrowth of its efforts to devise more systematic, objective, and precise bases for program decisions. The Division has determined, moreover, that the key to the success of its system is to insure that the Local Education Agency has an advanced planning capability.

Grateful acknowledgment is given to Dr. Robert M. Worthington, former Assistant Commissioner of Education (DVE), for initiating this study and to Mr. Stephen Poliacik, Assistant Commissioner of Education (DVE), for his guidance and support in continuing the study when problems seemed insurmountable. Also, to Former Commissioner of Education, Dr. Carl L. Marburger, and Acting Commissioner of Education, Dr. Edward W. Kilpatrick for their support and patience. Appreciation is further expressed to the Superintendents of the five LEAs: Mr. Charles A. Boyle, Edison; Mr. Americo R. Taranto, Linden; Mr. Jospeh R. Wilson, Somerset; Mr. Leonard A. Westman; Lower Camden County Regional High School; and Dr. J. Henry Zanzalari, Middlesex County Vocational Schools and Technical Institute for their cooperation and understanding.

Finally, to the staff of the Division of Vocational Education, and particularly Dr. Morton Margules, Associate Director, State Division of Vocational Education (Ancillary Services); Mr. Harold R. Seltzer, Director, Bureau of Occupational Research and Development; and Mr. Alvin Weitz, Director of Program Development for their invaluable assistance and insights. To Government Studies and Systems, Inc., Mr. Charles P. Cella, Director; Mr. Roger L. Sisson, Associate Director; Mr. Joseph H. Bosworth, Program Director; and Mr. Nelson G. Freed, Project Manager for their knowledge and technical capability so necessary in developing and testing this planning system.

The principal author of Volume III is:

Charles I. Goldman

Series Prefa

Planning is a universal concept based on the proposition that if you think a bit about what you intend to do, you are likely to do whatever it is better than if you don't think about it. This process of thinking ahead generally involves gathering information, analyzing the information and then formulating one or more courses of action to follow. The planning system presented here embodies these elements in operational procedures for planning for school districts.

The Vocational Education Planning System for Local School Districts draws heavily upon a growing body of experience in educational planning which has been generated by Government Studies and Systems (GSS). The introduction describes these concepts. Out of this experience has evolved a set of planning techniques, particularly suited by design and through actual use, to enable effective planning. The bases for the uses of indicators, planning factors, forecasts, models and others of these techniques are clearly laid out in this manual as they appear in the normal course of the planning cycle.

This manual is one of several resulting from a project to design planning procedures for local and state vocational education agencies. This manual describes the overall planning process for LEAs. It is to be used in conjunction with the following manuals:

Volume I : Local Education Agency Users' Manual

Volume II: Local Education Agency Users' Data Collection Manual

Volume III : Local Education Agency Planning Analyst's Procedures

Volume IV : State Application Funding Procedures

Volume V : Enrollment Forecasting Procedures

Volume VI : Procedure for Estimating Adult and Post-Secondary Potential Enrollment

Volume VII : Job Demand Forecasting Program

Volume VIII: Training Materials

Volume IX : Guide to Project Manuals

The most important ingredients in effective planning, however, are the people who do the planning. The planning team itself should include, at the very least, those who are going to be directly responsible for the execution of the plan, once developed, and those who are otherwise directly affected by the plan. People who participate in the planning process, who see their input take shape in a plan, tend to be better advocates and implementors of that plan.

TABLE OF CONTENTS

	Page
Acknowledgments	iii
Series Preface	v
Introduction	1
Section 1.0 Starting an ID on CSS and Loading the Permanent CSS Files for the Model	6
Section 2.0 Creating CSS Files of "Card Type" Input Data	7
Section 3.0 Error Checking and Editing of the CSS Files	9
Section 4.0 Start-up of a RAMIS Database	13
Section 5.0 Input of the "Card Type" Data into the RAMIS Database	17
Section 6.0 Playing Back the Input Data for Error Checking	31
Section 7.0 Error Checking if Non-Vocational Course Types and Programs are Used	36
Section 8.0 Input of the Top Level Data and Overhead Account Data	40
Section 9.0 Overall Error Checks Prior to Running the Model	48
Section 10.0 Running the Model	51
Section 11.0 Re-running the Model	90
Appendix - Annotated File Descriptions for FILE PROGRAMS, FILE SPACE, and FILE TMPCSE	93

INTRODUCTION

This manual is part of the supporting technical documentation for a vocational education planning process for local school districts, county vocational schools, and area vocational-technical schools. This manual details the steps required to process a local education agency's (LEA's) data after the data have been coded onto keypunch forms. Program, course, and overhead data are input into a computer data base and error checks are performed. A computer model is run using the data base to produce current year reports and five year projections of:

- Manpower needs met and student demand served
- Facilities needed versus available
- Teachers required, available and hires
- Program direct costs: teacher salary, materials and supplies, and travel
- Equipment and maintenance
- Total overhead costs by account and their allocations to programs.

The procedures for (1) input of the data, (2) performing error checks and (3) the modelling itself, are all written using the Rapid Access Management Information System (RAMIS) developed and supported by MATHEMATICA, Inc., Princeton, New Jersey.

RAMIS is a set of generalized data management and report generating computer software available on a lease/purchase arrangement from MATHEMATICA or surcharge arrangement with National CSS, Inc., Stamford, Connecticut, a time-shared and remote batch computer service.

This manual assumes the user knows the general procedures of RAMIS for input, updating, deleting and reporting data. All the specific RAMIS procedures for input of the vocational education data and the production of reports by the model for the planning process are contained in this manual and are recorded in associated computer files.

This manual assumes that the data will be input and the model run using the National CSS, Inc., computer service. The user should be familiar with the CSS procedures for creating, inputing and editing CSS data files and the CSS procedures for using RAMIS.

The following manuals should be referred to in using the procedures contained in this manual:

(1) For information about coding the data onto card forms, card formats, and sample output descriptions, refer to Volume II: Local Educational Agency User's Data Collection Manual (Division of Vocational Education, New Jersey State Department of Education).

(2) For information about RAMIS output, editing, and run procedures, refer to RAMIS Users Manual, and RAMIS Users Guide to the Run Executive (MATHEMATICA, Princeton, New Jersey).

(3) For information about National CSS procedures, refer to VP/CSS Reference Manual and VP/CSS Edit Command (National CSS, Inc., Stamford Connecticut).

This manual is organized in the following way:

	Sections
Information needed to set up the files, etc., initially	1 & 4
Procedures for data entry and checking at start of a cycle	2,3,5,6,7
Final data entry and edit	8,9
Initial run of the model in the cycle	10
Reruns of the model to examine alternatives (or after error correction)	11
File descriptions	Appendix

1.0 Starting an ID on CSS and Loading the Permanent CSS Files for the Model

1.1 Ask National CSS to create your ID and password with between 10 to 15 cylinders. After the ID is valid, create the PROFILE EXEC CSS file and set the permanent address.

1.2 Load from tape or offline punch to the ID the following CSS files:

STARTDB DATA

FILEDES DATA

INP1 DATA

PLYBTAB DATA

BIGM DATA

Listings and descriptions of these files along with files to be created during the processing of the LEA will be presented in later sections.

1.3 The processing of an LEA's data is done a level (secondary, post-secondary, or adult) at a time. The card form data is separated by level and separate RAMIS databases and runs of the model are made for each level.

2.0 Creating CSS Files of "Card Type" Input Data

After the card forms are completed, reviewed, and keypunched as described in Volume II: Local Educational Agency User's Data Collection Manual, the "card type" data for a level of the LEA are input to CSS files.

2.1 The following CSS files will be created:

CRD DATA

CRE DATA

CUR DATA

ENR DATA

EQP DATA

FAC DATA

MAN DATA

PGM DATA

PPR DATA

SOC DATA

Each file corresponds to a "card type" in Volume II: Local Educational Agency User's Data Collection Manual, and each file should contain all cards of that "card type" for the LEA-level. The use of the other "card types" will be described in section 8.0.

2.2 Before reading in the cards for an LEA-level, make sure that no files with these names exist for a prior LEA-level by typing "listf" in CSS mode to

print all CSS filenames or "listf crd data," etc. to check the prior existence of several of the files. If the files already exist, execute CSS tape dumps of the files and CSS "erase" the files.

2.3 Sort the cards by level by "card type" using a card sorter first on columns 2-4 and then on column 5. Separate out the "card type" data for the level of the LEA to be processed. Before each set of cards for a "card type," place a card with OFFLINE READ CRD DATA or OFFLINE READ CRE DATA, etc. Place an ID card before the whole deck of cards.

2.4 Send the cards to National CSS and have the cards read in. After the cards are read in, type "offline read" in CSS mode.

3.0 Error Checking and Editing of the CSS Files

Before input of the "card type" data to a RAMIS database, some error checks and edits may be performed on the CSS files.

3.1 The following "card types" are completed for programs and therefore should have the same number of cards in their CSS files: CUR, EQP, MAN, PGM, PPR, and SOC. Check the card counts by counting cards from listings of each file or by getting card counts under the ITEMS column of CSS "listf."

If academic and related programs are input (in addition to the vocational programs) there should be one card per academic and related program in the CUR, EQP, PGM, and PPR files. The MAN and SOC files will contain fewer cards (only cards for vocational programs).

3.2 Edit the CSS files to zero-fill key fields. Key fields are easier to work with when there are no leading or trailing blanks. The following columns should be zero-filled:

CSS file	Key field	Columns
CRD DATA	Course number	18-23
	Space type	32-34
CRE DATA	Program id	7-11
	Course number	18-23

CSS file	Key field	Columns
CUR DATA	Program id	7-11
ENR DATA	Program id	7-11
EQP DATA	Program id	7-11
FAC DATA	Space type	17-19
MAN DATA	Program id	7-11
PGM DATA	Program id	7-11
PPR DATA	Program id	7-11
SOC DATA	Program id	7-11

Missing program identification numbers, course codes, and space types will have all zeroes for the codes. Missing codes can then be identified on the error checking tables of section 5.0 as all zeros. Updating the missing codes in the RAMIS database can use the all zero codes to locate the records.

The program identification number should only be zero-filled for the vocational programs. Non-vocational programs like Mathematics, English, Science, Physical Education, etc., will have one character (left-justified) program identification numbers (M,E,S,P, etc.). The program identification numbers of the non-vocational programs should not be zero-filled because the identification numbers must later match non-vocational course types.

To zero-fill a key field, type "edit filename filetype." In edit mode, use "zone" to specify the columns and use "change." For example, to zero-fill course number on CRD cards, type the following:

```
edit crd data          (CSS mode)
z 18 23
c. / /0/ * *           (Edit mode)
file
```

To zero-fill program identification numbers where non-vocational programs are used, use "locate" or "next" to point to the start of a set of vocational program cards and use "change / /0/ n *" to zero-fill n consecutive vocational program identification numbers.

3.3 Each "card type" should be checked for the possibility that format field sizes were too small. List each CSS file using CSS "print" or "offline print." Scan down each listing to see if all cards of a file have the same number of columns punched.

If some cards of a file have more columns punched, field sizes might have been too small, and the LEA

had to exceed the field size. Edit the CSS file to make the columns conform and note the data changes which must be entered later because the assumed field sizes are exceeded. After the "card type" data are input using the procedures of section 5.0, enter the data changes using RAMIS SCAN or UPDATE.

4.0 Start-up of a RAMIS Database

Each LEA-level has its own database. The database resides in a CSS file called RAMDATA DATA. To begin the RAMIS processing of a new LEA-level, a new RAMDATA DATA must be created.

4.1 Make sure that no prior RAMDATA DATA CSS file exists on your ID. To test its existence, type "listf ramdata data" in the CSS mode. The system should respond with file not found. If RAMDATA DATA does exist, do a CSS tape dump and CSS erase of RAMDATA DATA or alter the name of the prior RAMDATA DATA.

4.2 In CSS mode, type "startup ramis 125." 125 tracks will be allocated for the database. If more tracks are needed, RAMIS will automatically add the tracks.

4.3 File descriptions for FILE PROGRAMS, FILE SPACE, and FILE TMPCSE must be input to the MASTER file of the new RAMDATA DATA database. (See Appendix for annotated file descriptions.) To input the file descriptions, in CSS mode type "ramisin startdb data." STARTDB DATA contains the following Ramis procedure. The file descriptions are found in file FILEDES DATA, whose listing follows.

p startdb data

```
INPUT
READ/FILEDES DATA
RECORD/48
FORM/12,4,4,12,12,4
ORDER/1,2,3,4,5,6
FILE MASTER
END
FINISH
```

14.59.54 p filedes data

PROGRAMS	1	4LEA-NUMBER	L	A6
PROGRAMS	1	4LEA-NAME	LN	A60
PROGRAMS	2	2CUR-FISC-YR	CFY	I4
PROGRAMS	2	2V-TCHR-ATTR	VTAR	F4.1
PROGRAMS	2	2N-TCHR-ATTR	NTAR	F4.1
PROGRAMS	2	2FRNG-RATE	FBR	F4.1
PROGRAMS	2	2SUBST-ALLOW	SAR	F4.1
PROGRAMS	2	2AVER-SAL-INCASII		F4.1
PROGRAMS	2	2ENTR-SAL-INCESII		F4.1
PROGRAMS	3	6PLAN-YEAR	YOP	I4
PROGRAMS	3	6VOC-EQUIP	VVE	I8
PROGRAMS	3	6TOT-SQFT	TSQFT	I8
PROGRAMS	3	6VOC-ADMIN-FTVASP		I8
PROGRAMS	3	6PCT-EQP-MNT	EMCR	F4.1
PROGRAMS	3	6TOTAL-STAFF	TSTAF	F7.1
PROGRAMS	3	6FTE-STUD	TFTS	F7.1
PROGRAMS	3	6VOC-ADM-STAFVASTF		F7.1
PROGRAMS	3	6MOOV-PER-STUMAIPP		F7.2
PROGRAMS	3	6VOC-INS-FT	VTCP	I8
PROGRAMS	3	6VOC-STUD	VSTU	F7.1
PROGRAMS	4	-19OVER-ACT-NO	OAOC	A5
PROGRAMS	4	-19OVER-ACT-NAMOAN		A20
PROGRAMS	4	-19OVER-INFL-RTOAIR		F4.1
PROGRAMS	4	-19OVER-BASE-TYOABST		A1
PROGRAMS	4	-19OVER-BASE-L	OABSL	F9.1
PROGRAMS	4	-19OVER-BASE-V	OABSV	F9.1
PROGRAMS	4	-19AMT-PER-BASEOAAPB		F8.2
PROGRAMS	4	-19OVER-AMT-L	OAMTL	F9.0
PROGRAMS	4	-19OVER-AMT-V	OAMTV	F9.0
PROGRAMS	5	2LEVEL	LV	A1
PROGRAMS	5	2WEEKS-OPEN	NWO	F4.1
PROGRAMS	5	2PDS-PER-WEEKNSP		F4.1
PROGRAMS	5	2OVER-WGT-FACOWF		F5.2
PROGRAMS	6	10PROG-IDEN	PID	A5
PROGRAMS	6	10PROG-STATUS	PSC	A4
PROGRAMS	6	10PROG-TYPE	PTYP	A1
PROGRAMS	6	10PROG-CAT	PCAT	A1
PROGRAMS	6	10MTLS-RATE	MCRI	F4.1
PROGRAMS	6	10TRAVEL-RATE	TRVR	F4.1
PROGRAMS	6	10PROG-NAME	PN	A25
PROGRAMS	6	10TCHR-REQ	NOTR	F5.1
PROGRAMS	6	10TCHR-AVL	NOTA	F5.1
PROGRAMS	6	10AVER-SAL	ASAL	I7
PROGRAMS	6	10ENT-SAL	ESAL	I6
PROGRAMS	6	10MTLS-COST-STMSCPS		F7.2
PROGRAMS	6	10TRAV-COST-STTCPS		F7.2
PROGRAMS	6	10COMP-RATE	CMPR	F5.1
PROGRAMS	6	10PLACE-RATE	PLER	F5.1
PROGRAMS	6	10COUN-MAN-FORCFMN		I6
PROGRAMS	6	10LEA-MAN-PCT	PLCF	F5.1
PROGRAMS	6	10AVER-CL-SIZEACSD		F5.1
PROGRAMS	6	10TCHR-PDS	TAVP	F4.1
PROGRAMS	6	10TCHR-WKS	TAVW	F4.1
PROGRAMS	6	10NEW-EQUIP	CNE	F8.0
PROGRAMS	6	10REPL-EQUIP	REC	F8.0
PROGRAMS	6	10SOC-DEMAND	SOCD	I6
PROGRAMS	6	10BASE-SALARY	BSAL	F8.0
PROGRAMS	6	10FRINGE-BENE	FRGE	F8.0
PROGRAMS	6	10SUBST-EXP	SUBE	F8.0

```
PROGRAMS      6  10MAT-AND-SUP MTSE        F8.0
PROGRAMS      6  10TRAVEL     TRVE        F8.0
PROGRAMS      6  10MAINTENCE  MAIE        F8.0
PROGRAMS      6  10OVERHEAD   OVRE        F8.0
PROGRAMS      7  -2REVENUE-CODERVSC        A5
PROGRAMS      7  -2REVENUE-NAMERVSN        A20
PROGRAMS      7  -2REV-PCT     PTPC        F5.1
PROGRAMS      8   3PROG-YR     PYR         I1
PROGRAMS      8   3FINAL-YEAR  FYON        A1
PROGRAMS      8   3STUD-IN-PROGPSTU        I8
PROGRAMS      9   3COURSE      CC          A6
PROGRAMS      9   3COURSE-TYPE CTY         A1
PROGRAMS      9   3COURSE-NAME CN          A20
PROGRAMS      9   3OE-CODE     OE          I8
PROGRAMS      9   3SPACE-TYPE  SPT         A3
PROGRAMS      9   3CRS-PDS-WK  PPW         F4.1
PROGRAMS      9   3CRS-LENGTH  CLT         F4.1
PROGRAMS     10   2GRADE       G           I2
PROGRAMS     10   2ENROLLMENT  CSTU        I4
SPACE         1   4LEA-NUMBER  L           A6
SPACE         1   4LEA-NAME    LN          A60
SPACE         2   2CUR-FISC-YR CFY         I4
SPACE         3   6PLAN-YEAR   YOP         I4
SPACE         4  10SPACE-TYPE  SPT         A3
SPACE         4  10SPACE-USE   SPU         A1
SPACE         4  10FAC-AVAIL   FAV         F5.1
SPACE         4  10STAT-AVAIL  STAV        F6.1
SPACE         4  10SQ-FT-PER-STSFPS        F5.1
SPACE         4  10PDS-AVL-WK  PPWA        F4.1
SPACE         4  10STAT-UTILIZ STUR        F5.1
SPACE         4  10SPACE-NAME  SPN         A25
SPACE         5   4LEVEL       LV          A1
SPACE         5   4WEEKS-OPEN  WKO         F4.1
SPACE         6   1PROG-IDEN   PID         A5
SPACE         6   1PROG-STATUS PSC         A4
SPACE         6   1PROG-NAME   PN          A25
SPACE         6   1AVER-CL-SIZEACSD        F4.1
SPACE         7   3PROG-YR     PYR         I1
SPACE         8   3COURSE-CODE CC          A6
SPACE         8   3TOT-STUD-PDSSPRDS       F9.1
TMPCSE        1 100CC          CC          A6
TMPCSE        2   6YOP         YOP         I4
TMPCSE        2   6CTY         CTY         A1
TMPCSE        2   6CN          CN          A20
TMPCSE        2   6OE          OE          I8
TMPCSE        2   6SPT         SPT         A3
TMPCSE        2   6PPW         PPW         F4.1
TMPCSE        2   6CLT         CLT         F4.1
TMPCSE        3   1LV          LV          A1
TMPCSE        4   1PID         PID         A5
TMPCSE        5   3PYR         PYR         I1
```

5.0 Input of the "Card Type" Data into the RAMIS Database

The "card type" data in the CSS files created and edited in sections 2.0 and 3.0 are now input into the RAMIS database. During the input procedure, RAMIS checks the data for incorrect program identification numbers, program years, and course codes.

5.1 The RAMIS procedures in the CSS file INP1 DATA will be executed to input the "card type" data. A listing of INP1 DATA follows. Edit the first card of INP1 DATA (the EXEC card) to change the LEA code (354810 in the listing) and the years (1973, etc.).

INP1 DATA is executed under control of the RAMIS Run Executive. (See RAMIS User's Guide to the Run Executive.) The Run Executive inserts the LEA code and years into the procedures of INP1 DATA prior to their execution.

5.2 Refer to the file descriptions in the Appendix to interpret the field numbers in the INSERT, ORDER, SET, and SCALE statements of the RAMIS procedures in INP1 DATA.

Procedures (1) input the CRE cards for all years to the RAMIS database. The program identification numbers, program years, course codes, etc. of the CRE

cards setup the tree structure of the PROGRAMS file. Later "card types" which are input to the PROGRAMS file must have corresponding program identification numbers and course codes or else the data cards are rejected by RAMIS due to mismatching keys.

Procedure (2) inputs the CUR cards for the current year. Scaling is assumed for required teachers (41), available teachers (42), materials cost per student (45), and travel cost per student (46).

Procedures (3) input the MAN cards for all years. Scaling is assumed for county manpower forecast (49).

Procedures (4) input the PPR cards for all years. Scaling is assumed for periods per week per teacher (52).

Procedures (5) input the EQP cards for all years. Procedures (6) input the SOC cards for all years. Procedures (7) input the PGM cards for all years. Procedures (8) input the ENR cards for all years. Because the ENR cards are recorded by grade for each program, it is necessary to add program students across grades (across ENR's) for each program. The update flag for program students is turned on so that RAMIS will add across ENR's for each program and then the flag is turned off.

Procedures (9) input the FAC cards for all years. The FAC cards setup the tree structure for the SPACE file. Scaling is assumed for square feet per student (9) and periods per week a facility is available (10).

Procedure (10) inputs the CRD cards to the TMPCSE file. Scaling is assumed for the periods per week course (7). The TMPCSE file is structured by course code. Procedure (11) spins off from the PROGRAMS file the program id's and program years occurring for each course code and places the program id's and program years under the appropriate course codes of the TMPCSE file.

Procedures (12) spin off the course description data by program id, program year, and course code so that the same course description data can be input for all the programs and program years which use the same course. Procedures (12) also inputs the course description data to the PROGRAMS file.

5.3 Execute the procedures of INP1 DATA by typing in CSS mode:

```
set core 512

attach ramis

filedef 04 dsk save ft04f001 p1 recfm fb lrecl
96 blksize 960

bramisin inp1 data
```

5.4 Review the print-out of the RAMIS processing of the procedures in INP1 DATA for possible reject messages. The following are reasons for rejects:

Procedure(s)	Reason for reject
(2) - CUR	Program id does not match any program id on CRE cards
(3) - MAN	Program id does not match any program id on CRE cards
(4) - PPR	Program id does not match any program id on CRE cards
(5) - EQP	Program id does not match any program id on CRE cards
(6) - SOC	Program id does not match any program id on CRE cards
(7) - PGM	Program id does not match any program id on CRE cards
(8) - ENR	Program id or program year does not match any program id or program year on CRE cards
(11) - Spin-off from PROGRAMS to TMPCSE	Course code on CRE card does not match any course code on CRD cards

Corrections to the RAMIS database because of rejects should wait until the "card type" input data is printed using the procedures of section 6.0. The error correction procedures also described in section 6.0 should be followed for the reject data.

```
INP1     DATA     P1    ID=GSS              04/17/73    09.25.58          PAGE
        VP/CSS    ---       NATIONAL CSS, INC. (STAMFORD DATA CENTER)

EXEC * 354810,1973,1974,1975,1976,1977,1978
INPUT
READ/CRE DATA
INSERT/1=&1,3=&2,10=&2
FORM/X4,1,1,5,1,X3,1,1,6,2,4
ORDER/30,36,34,37,67,71,70,77,78
FILE PROGRAMS
END
INCLUDE
READ/CRE DATA
INSERT/1=&1,3=&2,10=&3
FORM/X4,1,1,5,1,X3,1,1,6,2,X4,4
ORDER/30,36,34,37,67,71,70,77,78
SET/10,30,34,36,37,67,70,71,77,78
FILE PROGRAMS
END
INCLUDE
READ/CRE DATA
INSERT/1=&1,3=&2,10=&4
FORM/X4,1,1,5,1,X3,1,1,6,2,X8,4
ORDER/30,36,34,37,67,71,70,77,78
SET/10,30,34,36,37,67,70,71,77,78
FILE PROGRAMS
END
INCLUDE
READ/CRE DATA
INSERT/1=&1,3=&2,10=&5
FORM/X4,1,1,5,1,X3,1,1,6,2,X12,4
ORDER/30,36,34,37,67,71,70,77,78
SET/10,30,34,36,37,67,70,71,77,78
FILE PROGRAMS
END
INCLUDE
READ/CRE DATA
INSERT/1=&1,3=&2,10=&6
FORM/X4,1,1,5,1,X3,1,1,6,2,X16,4
ORDER/30,36,34,37,67,71,70,77,78
SET/10,30,34,36,37,67,70,71,77,78
FILE PROGRAMS
END
INCLUDE
READ/CRE DATA
INSERT/1=&1,3=&2,10=&7
FORM/X4,1,1,5,1,X3,1,1,6,2,X20,4
ORDER/30,36,34,37,67,71,70,77,78
SET/10,30,34,36,37,67,70,71,77,78
FILE PROGRAMS
END
UPDATE
READ/CUR DATA
INSERT/1=&1,3=&2,10=&2
FORM/X4,1,X1,5,X4,4,X-4,4,5,5,X5,5,5
ORDER/30,34,41,42,43,44,45,46
SET/41,42,43,44,45,46
SCALE/41=.1,42=.1,45=.01,46=.01
```

(1)

(2)

```
FILE PROGRAMS
END
UPDATE
READ/MAN DATA
INSERT/1=&1,3=&2,10=&2
FORM/X4,1,X1,5,X4,2,X10,2,X10,4,X20,2
SCALE/49=.1
ORDER/30,34,47,48,49,50
SET/47,48,49,50
FILE PROGRAMS
END
UPDATE
READ/MAN DATA
INSERT/1=&1,3=&2,10=&3
FORM/X4,1,X1,5,X4,X2,2,X8,X2,2,X8,X4,4,X16,X2,2
SCALE/49=.1
ORDER/30,34,47,48,49,50
SET/47,48,49,50
FILE PROGRAMS
END
UPDATE
READ/MAN DATA
INSERT/1=&1,3=&2,10=&4
FORM/X4,1,X1,5,X4,X4,2,X6,X4,2,X6,X8,4,X12,X4,2
SCALE/49=.1
ORDER/30,34,47,48,49,50
SET/47,48,49,50
FILE PROGRAMS
END
UPDATE
READ/MAN DATA
INSERT/1=&1,3=&2,10=&5
FORM/X4,1,X1,5,X4,X6,2,X4,X6,2,X4,X12,4,X8,X6,2
SCALE/49=.1
ORDER/30,34,47,48,49,50
SET/47,48,49,50
FILE PROGRAMS
END
UPDATE
READ/MAN DATA
INSERT/1=&1,3=&2,10=&6
FORM/X4,1,X1,5,X4,X8,2,X2,X8,2,X2,X16,4,X4,X8,2
SCALE/49=.1
ORDER/30,34,47,48,49,50
SET/47,48,49,50
FILE PROGRAMS
END
UPDATE
READ/MAN DATA
INSERT/1=&1,3=&2,10=&7
FORM/X4,1,X1,5,X4,X10,2,X10,2,X20,4,X10,2
SCALE/49=.1
ORDER/30,34,47,48,49,50
SET/47,48,49,50
FILE PROGRAMS
```

(3)

```
END
UPDATE
READ/PPR DATA
INSERT/1=&1,3=&2,10=&2
FORM/X4,1,X1,5,X10,3,X15,3,X15,2
ORDER/30,34,51,52,53
SET/51,52,53
SCALE/52=.1
FILE PROGRAMS
END
UPDATE
READ/PPR DATA
INSERT/1=&1,3=&2,10=&3
FORM/X4,1,X1,5,X10,X3,3,X12,X3,3,X12,X2,2
ORDER/30,34,51,52,53
SET/51,52,53
SCALE/52=.1
FILE PROGRAMS
END
UPDATE
READ/PPR DATA
INSERT/1=&1,3=&2,10=&4
FORM/X4,1,X1,5,X10,X6,3,X9,X6,3,X9,X4,2
ORDER/30,34,51,52,53
SET/51,52,53
SCALE/52=.1
FILE PROGRAMS
END
UPDATE
READ/PPR DATA
INSERT/1=&1,3=&2,10=&5
FORM/X4,1,X1,5,X10,X9,3,X6,X9,3,X6,X6,2
ORDER/30,34,51,52,53
SET/51,52,53
SCALE/52=.1
FILE PROGRAMS
END
UPDATE
READ/PPR DATA
INSERT/1=&1,3=&2,10=&6
FORM/X4,1,X1,5,X10,X12,3,X3,X12,3,X3,X8,2
ORDER/30,34,51,52,53
SET/51,52,53
SCALE/52=.1
FILE PROGRAMS
END
UPDATE
READ/PPR DATA
INSERT/1=&1,3=&2,10=&7
FORM/X4,1,X1,5,X10,X15,3,X15,3,X10,2
ORDER/30,34,51,52,53
SET/51,52,53
SCALE/52=.1
FILE PROGRAMS
END
```

(4)

```
UPDATE
READ/EQP DATA
INSERT/1=&1,3=&2,10=&2
FORM/X4,1,X1,5,X4,5,X25,5
ORDER/30,34,54,55
SET/54,55
FILE PROGRAMS
END
UPDATE
READ/EQP DATA
INSERT/1=&1,3=&2,10=&3
FORM/X4,1,X1,5,X4,X5,5,X20,X5,5
ORDER/30,34,54,55
SET/54,55
FILE PROGRAMS
END
UPDATE
READ/EQP DATA
INSERT/1=&1,3=&2,10=&4
FORM/X4,1,X1,5,X4,X10,5,X15,X10,5
ORDER/30,34,54,55
SET/54,55
FILE PROGRAMS
END
UPDATE
READ/EQP DATA
INSERT/1=&1,3=&2,10=&5
FORM/X4,1,X1,5,X4,X15,5,X10,X15,5
ORDER/30,34,54,55
SET/54,55
FILE PROGRAMS
END
UPDATE
READ/EQP DATA
INSERT/1=&1,3=&2,10=&6
FORM/X4,1,X1,5,X4,X20,5,X5,X20,5
ORDER/30,34,54,55
SET/54,55
FILE PROGRAMS
END
UPDATE
READ/EQP DATA
INSERT/1=&1,3=&2,10=&7
FORM/X4,1,X1,5,X4,X25,5,X25,5
ORDER/30,34,54,55
SET/54,55
FILE PROGRAMS
END
```

(5)

```
UPDATE
READ/SOC DATA
INSERT/1=&1,3=&2,10=&2
FORM/X4,1,X1,5,X10.4
ORDER/30,34,56
SET/56
FILE PROGRAMS
```

```
END
UPDATE
READ/SOC DATA
INSERT/1=&1,3=&2,10=&3
FORM/X4,1,X1,5,X10,X4,4
ORDER/30,34,56
SET/56
FILE PROGRAMS
END
UPDATE
READ/SOC DATA
INSERT/1=&1,3=&2,10=&4
FORM/X4,1,X1,5,X10,X8,4
ORDER/30,34,56
SET/56
FILE PROGRAMS
END
UPDATE
READ/SOC DATA
INSERT/1=&1,3=&2,10=&5
FORM/X4,1,X1,5,X10,X12,4
ORDER/30,34,56
SET/56
FILE PROGRAMS
END
UPDATE
READ/SOC DATA
INSERT/1=&1,3=&2,10=&6
FORM/X4,1,X1,5,X10,X16,4
ORDER/30,34,56
SET/56
FILE PROGRAMS
END
UPDATE
READ/SOC DATA
INSERT/1=&1,3=&2,10=&7
FORM/X4,1,X1,5,X10,X20,4
ORDER/30,34,56
SET/56
FILE PROGRAMS
END
```

(6)

```
UPDATE
READ/PGM DATA
INSERT/1=&1,3=&2,10=&2
FORM/X4,1,X1,5,X16,2,2,X24,25
ORDER/30,34,38,39,40
SET/38,39,40
FILE PROGRAMS
END
UPDATE
READ/PGM DATA
INSERT/1=&1,3=&2,10=&3
FORM/X4,1,X1,5,X16,2,2,X24,25
ORDER/30,34,38,39,40
SET/38,39,40
```

(7)

```
FILE PROGRAMS
END
UPDATE
READ/PGM DATA
INSERT/1=&1,3=&2,10=&4
FORM/X4,1,X1,5,X16,2,2,X24,25
ORDER/30,34,38,39,40
SET/38,39,40
FILE PROGRAMS
END
UPDATE
READ/PGM DATA
INSERT/1=&1,3=&2,10=&5
FORM/X4,1,X1,5,X16,2,2,X24,25
ORDER/30,34,38,39,40
SET/38,39,40
FILE PROGRAMS
END
UPDATE
READ/PGM DATA
INSERT/1=&1,3=&2,10=&6
FORM/X4,1,X1,5,X16,2,2,X24,25
ORDER/30,34,38,39,40
SET/38,39,40
FILE PROGRAMS
END
UPDATE
READ/PGM DATA
INSERT/1=&1,3=&2,10=&7
FORM/X4,1,X1,5,X16,2,2,X24,25
ORDER/30,34,38,39,40
SET/38,39,40
FILE PROGRAMS
END
UPDATE
FILE MASTER
1=PROGRAMS,5=PSTU,11=SET=1,$
END
UPDATE
READ/ENR DATA
INSERT/1=&1,3=&2,10=&2,68=N
FORM/X4,1,X1,5,X4,1,X9,4
ORDER/30,34,67,69
SET/68,69
FILE PROGRAMS
END
UPDATE
READ/ENR DATA
INSERT/1=&1,3=&2,10=&3,68=N
FORM/X4,1,X1,5,X4,1,X9,X4,4
ORDER/30,34,67,69
SET/68,69
FILE PROGRAMS
END
UPDATE
```

(8)

```
READ/ENR DATA
INSERT/1=&1,3=&2,10=&4,68=N
FORM/X4,1,X1,5,X4,1,X9,X8,4
ORDER/30,34,67,69
SET/68,69
FILE PROGRAMS
END
UPDATE
READ/ENR DATA
INSERT/1=&1,3=&2,10=&5,68=N
FORM/X4,1,X1,5,X4,1,X9,X12,4
ORDER/30,34,67,69
SET/68,69
FILE PROGRAMS
END
UPDATE
READ/ENR DATA
INSERT/1=&1,3=&2,10=&6,68=N
FORM/X4,1,X1,5,X4,1,X9,X16,4
ORDER/30,34,67,69
SET/68,69
FILE PROGRAMS
END
UPDATE
READ/ENR DATA
INSERT/1=&1,3=&2,10=&7,68=N
FORM/X4,1,X1,5,X4,1,X9,X20,4
ORDER/30,34,67,69
SET/68,69
FILE PROGRAMS
END
UPDATE
FILE MASTER
1=PROGRAMS,5=PSTU,11=SET=0,$
END
```

```
INPUT
READ/FAC DATA
INSERT/1=&1,3=&2,4=&2
FORM/X15,1,3,X1,3,4,4,3,X15,3,X3,25
ORDER/6,5,7,8,9,10,11,12
SCALE/9=.1,10=.1
FILE SPACE
END
INCLUDE
INSERT/1=&1,3=&2,4=&3
FORM/X15,1,3,X1,3,4,4,X3,3,X12,3,X3,25
ORDER/6,5,7,8,9,10,11,12
SCALE/9=.1,10=.1
READ/FAC DATA
SET/4,5,6,7,8,9,10,11,12
FILE SPACE
END
INCLUDE
INSERT/1=&1,3=&2,4=&4
FORM/X15,1,3,X1,3,4,4,X6,3,X9,3,X3,25
```

(9)

```
ORDER/6,5,7,8,9,10,11,12
SCALE/9=.1,10=.1
READ/FAC DATA
SET/4,5,6,7,8,9,10,11,12
FILE SPACE
END
INCLUDE
INSERT/1=&1,3=&2,4=&5
FORM/X15,1,3,X1,3,4,4,X9,3,X6,3,X3,25
ORDER/6,5,7,8,9,10,11,12
SCALE/9=.1,10=.1
READ/FAC DATA
SET/4,5,6,7,8,9,10,11,12
FILE SPACE
END
INCLUDE
INSERT/1=&1,3=&2,4=&6
FORM/X15,1,3,X1,3,4,4,X12,3,X3,3,X3,25
ORDER/6,5,7,8,9,10,11,12
SCALE/9=.1,10=.1
READ/FAC DATA
SET/4,5,6,7,8,9,10,11,12
FILE SPACE
END
INCLUDE
INSERT/1=&1,3=&2,4=&7
FORM/X15,1,3,X1,3,4,4,X15,3,3,X3,25
ORDER/6,5,7,8,9,10,11,12
SCALE/9=.1,10=.1
READ/FAC DATA
SET/4,5,6,7,8,9,10,11,12
FILE SPACE
END
```

```
ERASE
FILE TMPCSE
END
INPUT
READ/CRD DATA
INSERT/2=&2,8=X,9=YYYYY,10=0
FORM/X16,1,6,8,3,3,X15,2,X6,20
ORDER/3,1,5,6,7,8,4
SCALE/7=.1
FILE TMPCSE
END
```

(10)

```
TABLE
FILE PROGRAMS
PRINT YOP AND LV AND PID AND PYR AND SAVE
BY CC
IF PTYP IS V
IF L IS &1
IF YOP IS &2
END
INCLUDE
READ/SAVE FT04F001
RECORD/96
```

(11)

```
INP1     DATA     P1     ID=GSS              04/17/73     09.25.58          PAGE

FORM/6,4,1,5,1
ORDER/1,2,9,10,11
SET/9,10,11
FILE TMPCSE
END
-----------------------------------------------------------------------------
TABLE
FILE TMPCSE
PRINT CC AND CTY AND CN AND OE AND SPT AND PPW AND CLT AND SAVE
 BY LV BY PID BY PYR
IF LV IS-NOT X
IF YOP IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&1,3=&2,10=&2
FORM/1,5,1,6,1,20,8,3,4,4
ORDER/30,34,67,70,71,72,73,74,75,76
SET/71,72,73,74,75,76
FILE PROGRAMS
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&1,3=&2,10=&3
FORM/1,5,1,6,1,20,8,3,4,4
ORDER/30,34,67,70,71,72,73,74,75,76
SET/71,72,73,74,75,76
FILE PROGRAMS
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&1,3=&2,10=&4
FORM/1,5,1,6,1,20,8,3,4,4
ORDER/30,34,67,70,71,72,73,74,75,76
SET/71,72,73,74,75,76
FILE PROGRAMS
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&1,3=&2,10=&5
FORM/1,5,1,6,1,20,8,3,4,4
ORDER/30,34,67,70,71,72,73,74,75,76
SET/71,72,73,74,75,76
FILE PROGRAMS
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&1,3=&2,10=&6
FORM/1,5,1,6,1,20,8,3,4,4
ORDER/30,34,67,70,71,72,73,74,75,76
SET/71,72,73,74,75,76
```

(12)

```
FILE PROGRAMS
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&1,3=&2,10=&7
FORM/1,5,1,6,1,20,8,3,4,4
ORDER/30,34,67,70,71,72,73,74,75,76
SET/71,72,73,74,75,76
FILE PROGRAMS
END
ERASE
FILE TMPCSE
END
FINISH
END *
```

6.0 Playing Back the Input Data for Error Checking

After the above "card type" data have been input, the data should be displayed and checked.

6.1 RAMIS table procedures for displaying the input data are contained in the CSS file PLYBTAB DATA whose listing follows. Edit PLYBTAB DATA so the current year replaces 1972 in EXEC *. The RAMIS Run Executive will automatically insert the current year whenever &1 appears.

6.2 Execute PLYBTAB DATA by typing "ramisin plybtab data" in CSS mode, if the number of course grades is less than 300. If the number of course grades exceeds 300, type "set core 512" followed by "bramisin plybtab data" in CSS mode. Retrieve the output from National CSS.

6.3 Review the tables for (1) missing program identification numbers, course codes, and space types (all zeroes), (2) missing data (zeroes), (3) improper scaling, and (4) failure to follow assumed formats. Following the card formats in Volume II: Local Educational Agency User's Data Collection Manual, the input procedures in INP1 DATA assume scaling for some data elements. Check the following data elements for scaling errors:

TABLE	Title of Table	Data Elements
(1)	CUR CARDS	TCHR-REQ
		TCHR-AVL
		MTLS-COST-ST
		TRAV-COST-ST
(2)	MAN CARDS	COUN-MAN-FOR
(3)	PPR CARDS	TCHR-PDS
(8)	FAC CARDS	SQ-FT-PER-ST
		PDS-AVL-WK
(9)	FAC CARDS	PDS-AVL-WK
(11)	CRE AND CRD CARDS	CRS-PDS-WK

6.4 Compare the list of space types (SPACE-TYPE) in TABLE (9) with the space types in TABLE (10). If TABLE (9) is missing some space types found in TABLE (10) additional FAC cards are needed.

6.5 If the number of errors is relatively small for a "card type," corrections to the database can be made with RAMIS SCAN. If a "card type" is completely wrong, edit INP1 DATA using the PUT command to put into a new CSS file those RAMIS procedures of INP1 DATA which input the "card type." Edit the "card type" data or alter the input procedures to correspond to the way the cards are punched and execute the new CSS file.

If the ENR cards have to be re-input using procedures (8) of INP1 DATA to edit program students, then first execute the following RAMIS SCAN to zero-out program students.

```
SCAN
FILE PROGRAMS
L  1= lea code,$1
C.  69=SET=0,$*
END
```

It is necessary to zero-out program students because the ENR cards are input with the update flag on.

p plybtab data

```
EXEC * 1972
OFFLINE
-----------------------------------------------------------------
TABLE
HEADING
CUR CARDS
.
FILE PROGRAMS
PRINT NOTR AND NOTA AND ASAL AND ESAL AND MSCPS AND TCPS BY PID BY PTYP
IF YOP IS &1
END
-----------------------------------------------------------------
TABLE
HEADING
MAN CARDS
.
FILE PROGRAMS
WRITE CMPR AND PLER AND CFMN AND PLCF BY PID BY PTYP BY YOP
IF PTYP IS V
END
-----------------------------------------------------------------
TABLE
HEADING
PPR CARDS
.
FILE PROGRAMS
PRINT ACSD AND TAVP AND TAVW BY PID BY PTYP BY YOP
END
-----------------------------------------------------------------
TABLE
HEADING
EQP CARDS
.
FILE PROGRAMS
PRINT CNE AND REC BY PID BY PTYP BY YOP
END
-----------------------------------------------------------------
TABLE
HEADING
SOC CARDS
.
FILE PROGRAMS
WRITE SOCD ACROSS YOP BY PID BY PTYP
IF PTYP IS V
END
-----------------------------------------------------------------
TABLE
HEADING
PGM CARDS
.
FILE PROGRAMS
PRINT MCRI AND TRVR AND PH BY PID BY PTYP
IF YOP IS &1
END
-----------------------------------------------------------------
TABLE
HEADING
ENR CARDS - ENROLLMENTS SUMMED OVER GRADES
.
FILE PROGRAMS
WRITE PSTU ACROSS YOP BY PID BY PYR
IF PTYP IS V
END
-----------------------------------------------------------------
TABLE
HEADING
FAC CARDS
.
FILE SPACE
PRINT SPU AND SPT AND FAV AND STAV AND SFPS AND PTRA AND STUR AND SPH
IF YOP IS &1
```

(1) (2) (3) (4) (5) (6) (7)

```
TABLE
HEADING
FAC CARDS
.
FILE SPACE
WRITE PPWA ACROSS YOP BY SPT
END
TABLE
HEADING
FAC CARDS
.
FILE PROGRAMS
COUNT CC BY SPT
IF YOP IS &1
IF PTYP IS V
END
TABLE
HEADING
CRE AND CRD CARDS
.
FILE PROGRAMS
PRINT CTY AND CN AND OE AND SPT AND PPU AND CLT AND G AND CSTU
BY PID  BY PYR BY CC
IF PTYP IS V
IF YOP IS &1
END
FINISH
END *
```

(9)

(10)

(11)

7.0 Error Checking if Non-Vocational Course Types and Programs are Used

An LEA has the option of including in the course data the non-vocational courses of vocational students. If this option is used by the LEA, the non-vocational courses of vocational students should be classified into non-vocational subject areas like Mathematics, English, Science, Study Room, Physical Education, etc. The Vocational courses should have a course type of V (field 71 in PROGRAMS file). The non-vocational courses should have a course type associated with the subject area: M for Mathematics, E for English, etc.

There should be one program for each of these non-vocational subject areas. The program identification numbers (field 34 in PROGRAMS file) should be the same code as the course type: M for Mathematics, E for English, etc. The non-vocational programs should have a program type of either N for non-vocational or a related/academic classification may be used with non-vocational subject areas having a program type as either R for related or A for academic (field 36 in PROGRAMS file). Vocational programs should have program type V.

The non-vocational courses are input under the vocational porgrams in the tree structure of PROGRAMS file. The model spins-off the total course students periods by non-vocational course type and inputs these data under the non-vocational programs.

7.1 First check the correspondence between the non-vocational course types and the non-vocational programs. Execute the following RAMIS table:

```
TABLE
FILE PROGRAMS
COUNT CC BY CTY
IF PTYP IS V
IF CTY IS-NOT V
IF YOP IS current year
END
```

Compare the non-vocational course types with the non-vocational program identification numbers on table (1) (CUR cards) of Section 6.0. If the course types and program identification numbers do not agree, either the non-vocational course types have to be changed using RAMIS SCAN or additional non-vocational program data must be input.

If additional non-vocational programs are needed, input the following data to the PROGRAMS file using RAMIS INCLUDE, UPDATE, and SCAN for each non-vocational program:

Field Number	Description	Years
34	Program identification number (same as course type)	All years
36	Program type (N,R, or A)	All years
38	Materials rate (%)	All years
39	Travel rate (%)	All years
40	Program name (name of subject area)	All years
41	Teachers required	Current year
42	Teachers available	Current year
43	Average salary	Current year
44	Entering salary	Current year
45	Materials cost/full-time equivalent student	Current year
46	Travel cost/full-time equivalent student	Current year
51	Average class size	All years
52	Teacher periods per week	All years
53	Teacher weeks per year	All years
54	New equipment dollars	All years
55	Replacement equipment dollars	All years
67	Program year = 0	All years

As part of the error checking of section 6.0, the presence and format of the above data would have been checked for the non-vocational programs along with the vocational programs.

7.2 The model spins-off the total course student periods by non-vocational course type and includes these data under the non-vocational programs with year of program (field 67 in the PROGRAMS file) equal to 1. No year of program except 0 should exist for the non-vocational programs prior to running the model. As a check, execute the following RAMIS table:

```
TABLE
FILE PROGRAMS
PRINT PYR BY PID BY PTYP BY PN
IF PTYP IS-NOT V
IF YOP IS current year
END
```

If any year of program is printed other than 0, it must be deleted. Use RAMIS SCAN in the following way. Suppose the Mathematics program (M) had a year of program equal to 1. Then the SCAN would be:

```
SCAN
FILE PROGRAMS
SHOW 1,3,10,30,34,67
LOCATE 34=M, 67=1,$1
DELETE *
END
```

8.0 Input of the Top Level Data and Overhead Account Data

8.1 Make sure no prior CSS file called INP2 DATA exists for the ID by typing "listf inp2 data" in CSS mode. If a prior INP2 DATA exists, execute a tape dump and "erase inp2 data" in CSS mode.

8.2 Create INP2 DATA as shown in the following listing inserting the data for the LEA-level to be processed.

Procedure (1) makes sure the update flag for total vocational equipment is off and should be copied as is.

Procedure (2) inputs the highest level data. Refer to the card forms in Volume II: Local Educational Agency User's Data Collection Manual. Insert the LEA code after 1=, the LEA name after 2=, the current year after 3=, vocational teacher attrition percent (LPR card, Cols. 16-17) after 4=, non-vocational teacher attrition percent (LPR, Cols. 18-19) after 5=, fringe benefit percent (LPR, Cols. 20-21) after 6=, substitution percent (LPR, Cols. 22-23) after 7=, average salary percent change (LPR, Cols. 24-25) after 8=, and entering salary percent change (LPR, Cols. 26-27) after 9=.

Procedure (3) inputs data on the year of plan level. Change the LEA code (1=) and the current year (3=) for INSERT. After FILE PROGRAMS, insert data according to the following format:

Cols. 1-4	Year
Cols. 5-11	Vocational equipment dollars not including equipment under vocational programs (LPR, Cols. 40-47)
Cols. 12-17	Total LEA square feet (OVB, Cols. 46-53)
Cols. 18-22	Vocational administration square feet (OVB, Cols. 54-61)
Cols. 23-24	Maintenance cost as percent of equipment dollars (LPR, Cols. 28-39)
Cols. 25-29	Total LEA staff (OVA, Cols. 16-45)
Cols. 30-33	Total LEA full-time equivalent students (OVA, Cols. 46-75)
Cols. 34-37	Vocational administrative staff (OVB, Cols. 16-45)

Procedure (4) inputs level data. Change the LEA code (1=), the current year (3=), and the level code (30=) for INSERT. After FILE PROGRAMS, change the number of weeks open in Columns 5-6 and the total number of periods per week in columns 7-8, if necessary. Leave the overhead weighting factor at 1.0. The overhead weighing factor concept was not used in favor of a better approach toward allocating overhead.

Procedure (5) turns on the update flag for vocational equipment so that equipment under vocational programs may be added to current equipment. Leave

procedure (5) as is.

Procedures (6) and (7) input overhead account data. When a data base is first set up, each year of plan has one overhead account with blank account code (field 21 of the PROGRAMS file description) and blank or zero overhead data. Therefore, procedure (6) is an UPDATE to input one account for all years and the account code is used in SET. Procedure (7) then includes the rest of the accounts in all the years.

Change the LEA code (1=) and current year (3=) in the INSERT for both procedure (6) and procedure (7). The overhead account data to be input are assumed to be found in CSS files, OVH1 DATA and OVH2 DATA. Either type the data directly into the CSS files or edit previously used files or keypunch the overhead data and execute CSS OFFLINE READ.

The format of OVH1 DATA and OVH2 DATA is:

Cols. 1-4	Year of plan
Cols. 5-9	Overhead account code, left-justified
Cols. 10-29	Overhead account name
Cols. 30-33	Yearly percent change in overhead account cost per base (if overhead is input over time, leave blank)
Col. 34	Overhead account base for projection and allocation (S=

students, T=staff, Q=square feet). If the overhead account is not to be allocated to vocational programs, make the overhead account base X. If the account is not to be allocated, then the costs must be input over time. An account which is to be allocated (base = S, T, or Q) may be input over time).

Cols. 35-41 Overhead account cost (if the account is being projected, input the current year only and zero for later years)

Listings of example OVH1 DATA and OVH2 DATA follow. Note that OVH1 DATA is for one account for all years and OVH2 DATA is for the rest of the accounts for all years.

```
p inp2 data
```

```
UPDATE
FILE MASTER
1=PROGRAMS,5=VVE,11=SET=0,$
END
```

(1)

```
UPDATE
FILE PROGRAMS
1=233150,2=SET=MIDDLESEX COUNTY VOCATIONAL SCHOOLS,3=1972,
4=SET=10,5=SET=10,6=SET=15,7=SET=4.21,8=SET=10,9=SET=8,$
END
```

(2)

```
UPDATE
INSERT/1=233150,3=1972
FORM/4,7,6,5,2,5,4,4
ORDER/10,11,12,13,14,15,16,17
SET/11,12,13,14,15,16,17
FILE PROGRAMS
1972332637025716312044051 91.5191377.5
1973332637025716312044051 95.0210080.0
1974332637025716312044051 98.0220084.0
1975332637025716312044052 00.0240084.0
1976332637025716312044062 00.0260085.0
1977332637025716312044072 00.0280085.0
END
```

(3)

```
UPDATE
INSERT/1=233150,3=1972,30=S
FORM/4,2,2,3
ORDER/10,31,32,33
SET/31,32,33
FILE PROGRAMS
197238401.0
197338401.0
197438401.0
197538401.0
197638401.0
197738401.0
END
```

(4)

```
UPDATE
FILE MASTER
1=PROGRAMS,5=VVE,11=SET=1,$
END
```

(5)

```
UPDATE
READ/OVH1 DATA
INSERT/1=233150,3=1972
FORM/4,5,20,4,1,7
ORDER/10,21,22,23,24,28
SET/21,22,23,24,28
FILE PROGRAMS
END
```

(6)

```
INCLUDE
READ/OVH2 DATA
INSERT/1=233150,3=1972
FORM/4,5,20,4,1,7
ORDER/10,21,22,23,24,28
SET/21,22,23,24,28
FILE PROGRAMS
END
FINISH
```

(7)

```
14.32.51
```

```
p ovh1 data

19720100 ADMINISTRATION        08.6S 190652
19730100 ADMINISTRATION        08.6S      0
19740100 ADMINISTRATION        08.6S      0
19750100 ADMINISTRATION        08.6S      0
19760100 ADMINISTRATION        08.6S      0
19770100 ADMINISTRATION        08.6S      0
```

p ovn2 data

```
19720200AINSTRUCTION DIRECT   02.3S1812086
19730200AINSTRUCTION DIRECT   02.3S       0
19740200AINSTRUCTION DIRECT   02.3S       0
19750200AINSTRUCTION DIRECT   02.3S       0
19760200AINSTRUCTION DIRECT   02.3S       0
19770200AINSTRUCTION DIRECT   02.3S       0
19720200BINSTRUCTION INDIRECT02.9S  210120
19730200BINSTRUCTION INDIRECT02.9S       0
19740200BINSTRUCTION INDIRECT02.9S       0
19750200BINSTRUCTION INDIRECT02.9S       0
19760200BINSTRUCTION INDIRECT02.9S       0
19770200BINSTRUCTION INDIRECT02.9S       0
19720300 ATTENDANCE SERVICES  00.0S       0
19730300 ATTENDANCE SERVICES  00.0S       0
19740300 ATTENDANCE SERVICES  00.0S       0
19750300 ATTENDANCE SERVICES  00.0S       0
19760300 ATTENDANCE SERVICES  00.0S       0
19770300 ATTENDANCE SERVICES  00.0S       0
19720400 HEALTH SERVICES      01.3S   35980
19730400 HEALTH SERVICES      01.3S       0
19740400 HEALTH SERVICES      01.3S       0
19750400 HEALTH SERVICES      01.3S       0
19760400 HEALTH SERVICES      01.3S       0
19770400 HEALTH SERVICES      01.3S       0
19720500 TRANSPORTATION       25.3S    4000
19730500 TRANSPORTATION       25.3S       0
19740500 TRANSPORTATION       25.3S       0
19750500 TRANSPORTATION       25.3S       0
19760500 TRANSPORTATION       25.3S       0
19770500 TRANSPORTATION       25.3S       0
19720600 PLANT OPERATION      06.6Q  228693
19730600 PLANT OPERATION      06.6Q       0
19740600 PLANT OPERATION      06.6Q       0
19750600 PLANT OPERATION      06.6Q       0
19760600 PLANT OPERATION      06.6Q       0
19770600 PLANT OPERATION      06.6Q       0
19720700 PLANT MAINTENANCE    03.3Q  234264
19730700 PLANT MAINTENANCE    03.3Q       0
19740700 PLANT MAINTENANCE    03.3Q       0
19750700 PLANT MAINTENANCE    03.3Q       0
19760700 PLANT MAINTENANCE    03.3Q       0
19770700 PLANT MAINTENANCE    03.3Q       0
19720800AF.C. - STAFF         00.0T  226400
19730800AF.C. - STAFF         00.0T  175800
19740800AF.C. - STAFF         00.0T  184590
19750800AF.C. - STAFF         00.0T  193819
19760800AF.C. - STAFF         00.0T  203509
19770800AF.C. - STAFF         00.0T  213684
19720800BF.C. - STUDENT       00.0S       0
19730800BF.C. - STUDENT       00.0S       0
19740800BF.C. - STUDENT       00.0S       0
19750800BF.C. - STUDENT       00.0S       0
19760800BF.C. - STUDENT       00.0S       0
19770800BF.C. - STUDENT       00.0S       0
19720800CF.C. - PROPERTY      16.7Q   30000
19730800CF.C. - PROPERTY      16.7Q       0
19740800CF.C. - PROPERTY      16.7Q       0
19750800CF.C. - PROPERTY      16.7Q       0
19760800CF.C. - PROPERTY      16.7Q       0
19770800CF.C. - PROPERTY      16.7Q       0
19720900 FOOD SERVICES        09.3S   50000
19730900 FOOD SERVICES        09.3S       0
```

19750900	FOOD SERVICES	09.3S	0
19760900	FOOD SERVICES	09.3S	0
19770900	FOOD SERVICES	09.3S	0
19721000	STUDENT ACTIVITIES	30.2S	33600
19731000	STUDENT ACTIVITIES	30.2S	0
19741000	STUDENT ACTIVITIES	30.2S	0
19751000	STUDENT ACTIVITIES	30.2S	0
19761000	STUDENT ACTIVITIES	30.2S	0
19771000	STUDENT ACTIVITIES	30.2S	0
19721100	COMMUNITY SERVICES	00.0S	0
19731100	COMMUNITY SERVICES	00.0S	0
19741100	COMMUNITY SERVICES	00.0S	0
19751100	COMMUNITY SERVICES	00.0S	0
19761100	COMMUNITY SERVICES	00.0S	0
19771100	COMMUNITY SERVICES	00.0S	0
19721200	CAPITAL OUTLAY	00.00	125000
19731200	CAPITAL OUTLAY	00.00	120000
19741200	CAPITAL OUTLAY	00.00	122000
19751200	CAPITAL OUTLAY	00.00	125000
19761200	CAPITAL OUTLAY	00.00	130000
19771200	CAPITAL OUTLAY	00.00	132000
19721320	DEBT PRINCIPAL	00.00	0
19731320	DEBT PRINCIPAL	00.00	0
19741320	DEBT PRINCIPAL	00.00	0
19751320	DEBT PRINCIPAL	00.00	0
19761320	DEBT PRINCIPAL	00.00	0
19771320	DEBT PRINCIPAL	00.00	0
19721330	SERVICE INTEREST	00.00	0
19731330	SERVICE INTEREST	00.00	0
19741330	SERVICE INTEREST	00.00	0
19751330	SERVICE INTEREST	00.00	0
19761330	SERVICE INTEREST	00.00	0
19771330	SERVICE INTEREST	00.00	0

9.0 Overall Error Checks Prior to Running The Model

The presence of two overall inconsistencies in the data causes the allocation of overhead costs to vocational programs to be done incorrectly. Whether the inconsistencies are present must be determined and corrective procedures taken.

9.1 Check whether the total vocational students across programs exceeds total full-time students. Execute the following RAMIS table:

```
TABLE
FILE PROGRAMS
WRITE FST*TFTS AND PSTU BY YØP
IF PTYP IS V
END
```

If the total vocational students (PSTU) exceeds total full-time equivalent students (TFTS), make the total full-time equivalent students (field 16 in PROGRAM file) equal to total vocational students, using RAMIS SCAN or UPDATE.

9.2 Check whether total student periods under vocational programs (possibly including related and academic courses) exceeds total student periods. Execute the following two sets of RAMIS procedures.

Procedure 1:

```
DEFINE
FILE PROGRAMS
TOT-PERIODS/F12.0 = TFTS * NWO * NSP
END
TABLE
FILE PROGRAMS
WRITE TOT-PERIODS BY YOP
END
```

Procedure 2:

```
DEFINE
FILE PROGRAMS
VOC-PERIODS/F12.0 = CSTU * CLT * PPW
END
TABLE
FILE PROGRAMS
WRITE VOC-PERIODS BY YOP
IF PTYP IS V
END
```

If the total periods of vocational students (VOC-PERIODS) is less than or equal to total periods (TOT-PERIODS), no inconsistency is present. If the VOC-PERIODS exceeds TOT-PERIODS, then either (1) total students (TFTS) must be increased, (2) the number weeks open or the number

periods per week (NWO or NSP) are wrong, (3) or double counting has the course data wrong. Review the data and edit.

10.0 Running The Model

After all the "card type" data and top level data have been input and checked, the model is run as a batch job to be printed off-line.

10.1 The model is contained in the CSS file called BIGM DATA whose listing follows. The model consists of a set of RAMIS procedures to be executed under control of the RAMIS Run Executive. The Run Executive is used to insert parameter values like LEA code and to branch depending upon option flags (e.g., whether or not there are related/academic programs).

The RAMIS procedures have been grouped into sets of procedures, and each set of procedures has been annotated. Refer to the file descriptions in the Appendix for translation of the field numbers used in INSERT, ORDER, and SET statements. Refer to RAMIS User's Guide to the Run Executive for information on the Run Executive features used. Section 10.2 describes the first three cards and the &1 through &12 found in the listing.

Annotations of Sets of RAMIS Procedures in the Model

(1) Output the course and program descriptions reports.

(2) Output the vocational program enrollments reports.

(3) Output the course enrollments report and count of number of courses with non-zero enrollment report.

(4) Compute and store which program year is the last program year for each vocational program.

(5) Output the manpower needs fulfilled reports for vocational programs.

(6) Output the social demand met reports for vocational programs.

(7) Compute and store current year class-size for vocational programs.

(8) If there are related/academic courses, compute and store the total student-periods by related/academic course (program) types, e.g., Mathematics, English, Science, etc.

(9) Compute number of teachers required and hold.

(10) Compute number of teachers required, rounded to nearest .5, and store.

(11) Compute number of teachers available, rounded to nearest .5, and store.

Annotations of Sets of RAMIS Procedures in the Model

(12) Output teaching requirements report.

(13) Compute student-periods and store in SPACE file by space type by program by program year by course. Note use of INCLUDE.

(14) Spin-off number of weeks and class-sizes and store in SPACE file.

(15) Put program names in SPACE file.

(16) Output facilities and stations required versus available reports.

(17) Compute and store average salary and entering salary over time using inflation rates.

(18) Compute and store base salary cost, fringe benefit cost, and substitution cost.

(19) Output teaching expense reports.

(20) Compute and store materials cost per student and travel cost per student over time using inflation rates.

(21) If there are related/academic courses, compute and store full-time equivalent students in the related/academic programs as the basis for computing materials cost and travel cost for related/academic programs.

(22) Compute and store materials cost and travel cost.

(23) Output materials cost and travel cost reports.

Annotations of Sets of RAMIS Procedures in the Model

(24) Compute and store total vocational equipment.

(25) Compute total equipment maintenance and maintenance cost per vocational student and hold.

(26) Output total vocational equipment and maintenance report.

(27) Compute and store maintenance cost for each vocational program.

(28) Output the report for each vocational program on total program cost without overhead.

(29) Compute and store total vocational students and store total vocational students as the vocational base for each overhead account that uses students as its allocation base.

(30) Output total program cost without overhead.

(31) Store total full-time students as the LEA base in each overhead account that uses students as its allocation base.

(32) Store total square feet as the LEA base in each overhead account that uses square feet as its allocation base.

(33) Store total staff as the LEA base in each overhead account that uses staff as its allocation base.

(34) Change overhead base code to X for up to three overhead accounts which are not to projected over time but which are to be allocated to vocational programs.

Annotations of Sets of RAMIS Procedures in the Model

(35) Output the overhead account cost data which were input.

(36) Compute and store total instruction direct costs (account 0200A) less total vocational program and related/academic program instruction costs. Compute and store total maintenance cost (account 700) less vocational maintenance and replacement equipment. Compute and store total capital outlay (account 1200) less new equipment. All computations are for the current year.

(37) Output overhead account costs less current year vocational and related/academic program costs.

(38) Compute and store current year overhead account cost per LEA base for all overhead accounts that are to be projected over time.

(39) Store current year overhead account costs per LEA base in each of the future years for inflation in procedures (40)

(40) Compute and store (1) overhead account costs per LEA base using inflation percents and (2) LEA overhead account cost in each of the future years.

(41) Output overhead account costs per LEA base.

(42) If any overhead accounts are not to be projected over time but are to be allocated to vocational programs, restore their overhead base codes (S, Q, or T) which were set to X in procedures (34)

Annotations of Sets of RAMIS Procedures in the Model

(43) Output the LEA overhead account costs less vocational and related/academic costs.

(44) Compute and store total vocational staff and store as vocational base for all overhead accounts that use staff as the allocation base (T).

(45) Compute and store total vocational space and store as vocational base for all overhead accounts that use space as the allocation base (Q).

(46) Compute and store total student periods using full-time equivalent students and store in LEA base for account 0200A (Instruction Direct). In procedures (48), total course periods of vocational students will be deducted from total student periods to compute LEA base for account 0200A.

(47) Compute total vocational student periods using total vocational students and store in vocational base for account 0200A. In procedures (48), total course periods of vocational students will be deducted from total vocational student periods to compute vocational base for account 0200A.

(48) Compute total course periods of vocational students and deduct from total student periods in LEA base and total vocational student periods in vocational base for account 0200A and store.

Annotations of Sets of RAMIS Procedures in the Model

(49) If all the students in the LEA-level are not vocational students, then in procedure (51) allocate the vocational share of each overhead account using the ratio of vocational base to LEA base for each account.

(50) If all the students in the LEA-level are vocational students, then in procedure (51) allocate all the LEA overhead account cost.

(51) Compute and store the vocational amount of each overhead account cost that is to be allocated.

(52) Reverse the process of procedures (36) to add vocational and related/academic program costs to the appropriate LEA overhead account costs.

(53) Output total LEA overhead account costs with vocational and related/academic program costs.

(54) Compute total vocational overhead cost per student and store temporarily in maintenance cost per student.

(55) Compute and store the overhead allocation to each vocational program.

(56) Output the total vocational and related/academic program costs with overhead.

(57) Output the LEA and vocational overhead account bases: student periods (for account 0200A), student, staff, and space in square feet.

```
HIG4     DATA     P1     ID=GSS                    04/13/73    11.17.30          PAGE
         VP/CSS   ---          NATIONAL CSS, INC. (STAMFORD DATA CENTER)

EXEC * 'MIDDLESEX CO. VOC.-TECH. SCHOOLS - SECONDARY - MARCH 1973'.
233150.1972.1.1.2.08000.T.1200.0
&&ARGS 7=00000.8=Y.9=00000.10=Y.11=00000.12=Y
OFFLINE
TABLE
HEADING
&1
PROGRAM DESCRIPTION - PERIODS PER WEEK BY COURSE
.
FILE PROGRAMS
WRITE PPW ACROSS YOP
BY LV BY PID BY PYR BY CC BY CTY BY DE BY CN
IF L IS &2
IF RTYP IS V
END
RETYPE
TABLE
HEADING
&1
PROGRAM DESCRIPTION - COURSE LENGTH IN WEEKS BY COURSE
.
FILE PROGRAMS
WRITE CLT ACROSS YOP
BY LV BY PID BY PYR BY CC BY CTY BY DE BY CN
IF L IS &2
IF RTYP IS V
END
RETYPE
TABLE
HEADING
&1
PROGRAM DESCRIPTION - PERIODS PER WEEK BY PROGRAM
.
FILE PROGRAMS
WRITE PPW ACROSS YOP BY LV BY PID BY PC BY PYR
IF L IS &2
IF RTYP IS V
END
RETYPE
TABLE
HEADING
&1
PLANNED PROGRAM ENROLLMENT - BY PROGRAM
.
FILE PROGRAMS
WRITE RSTU AND COLUMN-TOTAL ACROSS YOP BY LV BY PID BY PN
IF L IS &2
IF RTYP IS V
END
RETYPE
TABLE
FILE PROGRAMS
WRITE RSTU AND HOLD BY YOP
IF L IS &2
IF RTYP IS V
```

```
END
DEFINE
FILE HOLD
LSTPSTU/I8=LSTPSTU
DIFPSTU/I8=E02-LSTPSTU
PCT-CHANGE/F5.1=.IF.E01.EQ.0.0.THEN.0.0.ELSE.(DIFPSTU/LSTPSTU)*100.0
LSTPSTU=E02
END
TABLE
HEADING
&1
PERCENT CHANGE IN TOTAL PROGRAM ENROLLMENT
.
FILE HOLD
PRINT E01 AND E02 AND PCT-CHANGE
END
RETYPE
TABLE
HEADING
&1
PLANNED PROGRAM ENROLLMENT - BY PROGRAM YEAR
.
FILE PROGRAMS
WRITE PSTU AND COLUMN-TOTAL ACROSS YOP BY LV BY PYR
IF L IS &2
IF PTYP IS V
END
RETYPE
TABLE
HEADING
&1
PLANNED COURSE ENROLLMENT
.
FILE PROGRAMS
WRITE CSTU ACROSS YOP
 BY LV BY PID BY PYR BY CC BY CTY BY DP BY CN BY CS
IF L IS &2
IF PTYP IS V
END
RETYPE
DEFINE
FILE PROGRAMS
CRSE-COUNT/I4=.IF.CSTU.GT.0.THEN.1.ELSE.0
END
TABLE
HEADING
&1
COUNT NUMBER OF VOCATIONAL COURSES
.
FILE PROGRAMS
WRITE CRSE-COUNT BY YOP
IF L IS &2
IF PTYP IS V
IF CTY IS V
END
```

(3)

```
RETYPE
TABLE
FILE PROGRAMS
WRITE MAX*PYR AND SAVE BY LV BY PID BY YOP
IF L IS &2
IF PTYP IS V
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.6R=Y
FORM/1.5.4.1
ORDER/30.34.10.67
SET/&8
FILE PROGRAMS
END
```

(4)

```
DEFINE
FILE PROGRAMS
LEA-REQT/F8.1=CFMN*(PLCE/100.0)
STUD-COMP/F8.1=PSTU*(COMPR/100.0)
STUD-PLCD/F8.1=PSTU*(COMPR/100.0)*(PLER/100.0)
STD-IN-LSTYR/I8=PSTU
END
TABLE
HEADING
&1
MANPOWER NEEDS FULFILLED
.
FILE OUTPUT
PCT-ND-MT/F5.1=(C05/C02)*100.0
RUN
FILE PROGRAMS
WRITE CFMN AND LEA-REQT AND STD-IN-LSTYR AND STUD-COMP AND STUD-PLCD
BY LV BY PID BY PN BY YOP
IF FYON IS Y
IF L IS &2
IF PTYP IS V
END
RETYPE
TABLE
HEADING
&1
MANPOWER NEEDS FULFILLED
.
FILE OUTPUT
LEA-MAN-PCT/F5.1=(C02/C01)*100.0
LEA-COMP-PCT/F5.1=(C04/C03)*100.0
LEA-PLAC-PCT/F5.1=(C05/C04)*100.0
PCT-ND-MT/F5.1=(C05/C02)*100.0
RUN
FILE PROGRAMS
WRITE CFMN AND LEA-REQT AND STD-IN-LSTYR AND STUD-COMP AND STUD-PLCD
BY YOP
IF FYON IS Y
IF CFMN EXCEEDS 0.1
```

(5)

```
IF L IS &2
IF PTYP IS V
END
RETYPE
TABLE
HEADING
&1
MANPOWER NEEDS FULFILLED - SUBSIDIARY DATA
.
FILE PROGRAMS
WRITE PLCF AND CMPR AND PLFR
BY LV BY PID BY PN BY YOP
IF FYON IS Y
IF L IS &2
IF PTYP IS V
END
RETYPE
-------------------------------------------------------------------
DEFINE
FILE PROGRAMS
STD-IN-FSTYR/I4=PSTU
END
TABLE
HEADING
&1
SOCIAL DEMAND ANALYSIS
.
FILE OUTPUT
PCT-DMD-SVD/F5.1=(C02/C01)*100.0
RUN
FILE PROGRAMS
WRITE SOCD AND STD-IN-FSTYR BY LV BY PID BY PN BY YOP
IF PYR IS 1
IF L IS &2
IF PTYP IS V
END
RETYPE
TABLE
HEADING
&1
SOCIAL DEMAND ANALYSIS
.
FILE OUTPUT
PCT-DMD-SVD/F5.1=(C02/C01)*100.0
RUN
FILE PROGRAMS
WRITE SOCD AND STD-IN-FSTYR BY YOP
IF PYR IS 1
IF L IS &2
IF PTYP IS V
END
RETYPE
-------------------------------------------------------------------
DEFINE
FILE PROGRAMS
CCS/F5.1=.IF.NOTR.EQ.0.0.THEN.0.0.
         ELSE.(CSTU*SJ**CLT)/(NOTR*TAVP*TAVR)
```

(6)

```
END
TABLE
FILE PROGRAMS
WRITE CCS AND SAVE
BY LV BY PID
IF YOP IS &3
IF L IS &2
IF PTYP IS V
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.10=&3
FORM/1.5.5
ORDER/30.34.51
SET/51
FILE PROGRAMS
END
&& IF &4 LT 1 THEN GOTO LAMBL1;
DEFINE
FILE PROGRAMS
SPRDS/F9.1=CSTU*PPW*CLT
END
TABLE
FILE PROGRAMS
WRITE SPRDS AND SAVE BY LV BY CTY BY YOP
IF YOP IS-NOT &3
IF L IS &2
IF PTYP IS V
IF CTY IS-NOT V
END
INCLUDE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.67=1.71=V.76=1.0.77=10.78=1
FORM/1.1.X-1.1.4.9
ORDER/30.34.70.10.75
SET/57
FILE PROGRAMS
END
&&LAMBL1
DEFINE
FILE PROGRAMS
TCHRSR/F8.2=.IF.TAVP.EQ.0.0.THEN.0.0.
            ELSE.((CSTU/ACSU)*PPW*CLT)/(TAV**TAVW)
END
TABLE
FILE PROGRAMS
WRITE TCHRSR AND HOLD
BY LV BY PID BY YOP
IF YOP IS-NOT &3
IF L IS &2
IF CTY IS V
IF PYR EXCEEDS 0
END
```

(7)

(8)

(9)

```
DEFINE
FILE HOLD
TCHRND/FR.2=INT(2.0**FO4 + .5)/2
END
TABLE
FILE HOLD
PRINT E01 AND E02 AND E03 AND TCHRND AND SAVE
END
UPDATE
HEAD/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/1.X3.5.X3.4.H
ORDER/30.34.10.41
SET/41
FILE PROGRAMS
END
DEFINE
FILE PROGRAMS
IMPR/F4.1=.IF.PTYP.EQ.'V'.THEN.VTAM.ELSE.NTAM
TCHRSA/FR.2=NOTRR(1.0-(T*PR/100.0))
TCHAND/FR.2=INT(2.**TCHRSA + .5)/2
AYOP/I4=YOP+1
LYOP/I4=CFY + 5
FLAG/I1=.IF.YOP.LT.LYOP.THEN.1.ELSE.
END
TABLE
FILE PROGRAMS
PRINT AYOP AND TCHAND AND SAVE
BY LV BY PID BY YOP
IF FLAG IS 1
IF L IS &2
END
UPDATE
HEAD/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/1.5.X4.4.H
ORDER/30.34.10.42
SET/42
FILE PROGRAMS
END
TABLE
HEADING
&1
TEACHING REQUIREMENTS - HOURLY TO [illegible]
.
FILE OUTPUT
TCHR-HIRES/F5.1=C04-C05
RUN
FILE PROGRAMS
PRINT ACSD AND TAVP AND TAVN AND NOT- AND PDTA
BY LV BY PID BY PN BY YOP
IF L IS &2
END
```

(10) (11) (12)

```
RETYPE
OFFLINE
FILE PROGRAMS
STUD-PROS/F9.1=CSTU>PRWSCLT
END
TABLE
FILE PROGRAMS
WRITE STUD-PROS AND SAVE
BY YOP BY LV BY PID BY PYR BY CC BY SPT
IF SPT IS-NOT 000
IF L IS &2
IF PTYP IS V
END
INCLUDE
READ/SAVE FT04F001
RECORD/9A
INSERT/1=&2.3=&3
FORM/4.1.5.1.6.3.9
ORDER/4.13.15.19.20.5.21
SET/13.15.19.20.21
FILE SPACE
END
```

(13)

```
TABLE
FILE PROGRAMS
PRINT N&O AND ACSD AND SAVE
BY YOP BY LV BY PID BY SPT
IF SPT IS-NOT 000
IF L IS &2
IF PTYP IS V
END
UPDATE
READ/SAVE FT04F001
RECORD/9A
INSERT/1=&2.3=&3
FORM/4.1.5.3.4.5
ORDER/4.13.15.5.14.1
SET/14.18
FILE SPACE
END
```

(14)

```
TABLE
FILE PROGRAMS
PRINT PN AND SAVE
BY YOP BY LV BY PID BY SPT
IF SPT IS-NOT 000
IF L IS &2
IF PTYP IS V
END
UPDATE
READ/SAVE FT04F001
RECORD/9A
INSERT/1=&2.3=&3
FORM/4.1.5.3.25
ORDER/4.13.15.5.17
SET/17
FILE SPACE
```

(15)

```
END
DEFINE
FILE SPACE
STATREQ/FS.1=.IF.PPWA.GT.0.0.THEN.SPWDS/(NWD*PPWA*(STUP/100.0)).ELSE.0.0
FACREQ/FS.1=.IF.ACSD.GT.0.0.THEN.STATREQ/ACSD.ELSE.0.0
SOFTREQ/FS.1=STATREQ*SFWS
END
TABLE
HEADING
A1
FACILITIES STATUS
.
FILE SPACE
WRITE FST*FAV AND FST*STAV AND FST*STUP AND STORE
BY SPU BY SPT BY SPN BY YOP
IF LV IS-NOT ** $*
IF L IS A2
RUN
FILE SPACE
WRITE FACREQ AND STATREQ
BY SPU BY SPT BY SPN BY YOP
IF LV IS-NOT ** $*
IF L IS A2
END
RETYPE
TABLE
HEADING
A1
FACILITIES STATUS - OVERALL
.
FILE SPACE
WRITE FAV AND STAV
BY SPU BY YOP
IF L IS A2
END
RETYPE
TABLE
HEADING
A1
STATIONS REQUIRED (BY PROGRAM)
.
FILE SPACE
WRITE FACREQ AND STATREQ AND SOFTREQ
BY LV BY PID BY PN BY SPT BY SPU BY YOP
IF LV IS-NOT ** $*
IF L IS A2
END
RETYPE
TABLE
FILE SPACE
WRITE FACREQ AND STATREQ AND HOLD
BY SPT BY SPU BY YOP
IF LV IS-NOT ** $*
IF L IS A2
END
```

(16)

```
HIGH     DATA     P1     ID=GSS                    04/13/83     11.17.30          PAGE
DEFINE
FILE HOLD
FACREQR/F6.1=INT(2.0*E04 + .5)/2
END
TABLE
HEADING
A1
FACILITIES STATUS - OVERALL (FAC. REQ. ROUNDED TO NEAREST .5 AND ADDED)
.
FILE HOLD
WRITE FACREQR AND E05 BY E02 BY E03
END
RETYPE
DEFINE
FILE SPACE
STA-PRDS/F10.1=STAV*PRNA*NWO
END
TABLE
FILE SPACE
WRITE FST*STA-PRDS AND SPRDS AND HOLD
BY SPT BY SPU BY SPN BY YOP
IF LV IS-NOT '*' $'
IF L IS A2
END
TABLE
HEADING
A1
FACILITIES UTILIZATION - BY SPACE TYPE
.
FILE OUTPUT
PCT-STA-UTIL/F5.1=(C02/C01)*100.0
RUN
FILE HOLD
PRINT E05 AND E06
BY E01 BY E02 BY E03 BY E04
END
RETYPE
TABLE
HEADING
A1
FACILITIES UTILIZATION - OVERALL
.
FILE OUTPUT
PCT-STA-UTIL/F5.1=(C02/C01)*100.0
RUN
FILE HOLD
WRITE E05 AND E06
BY E04
END
RETYPE
-------------------------------------------------------------------------------
DEFINE
FILE PROGRAMS
CFY1/I4=CFY+1
CFY2/I4=CFY+2
CFY3/I4=CFY+3
```

```
CFY4/I4=CFY+4
CFY5/I4=CFY+5
END
TABLE
FILE PROGRAMS
PRINT CFY1 AND ASAL AND ESAL AND CFY2 AND ASAL AND ESAL AND
      CFY3 AND ASAL AND ESAL AND CFY4 AND ASAL AND ESAL AND
      CFY5 AND ASAL AND ESAL AND SAVE
BY LV BY PID
IF YOP IS &3
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/1.5.4.7.6.C4.C7.C6.C4.C7.C6.C4.C7.C6.C4.C7.C6
ORDER/30.34.10.43.44.10.43.44.10.43.44.10.43.44.10.43.44
SET/43.44
FILE PROGRAMS
END
DEFINE
FILE PROGRAMS
AS/I7=ASAL*(1.0+(ASII/100.0))**(YOP-CFY)
ES/I6=ESAL*(1.0+(ESII/100.0))**(YOP-CFY)
END
TABLE
FILE PROGRAMS
PRINT AS AND ES AND SAVE
BY LV BY PID BY YOP
IF YOP IS-NOT &3
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/1.5.4.7.6
ORDER/30.34.10.43.44
SET/43.44
FILE PROGRAMS
END
```

(17)

```
DEFINE
FILE PROGRAMS
X/F8.2=NOTR-NOTA
TCHRHIRES/F8.2=.IF.X.GE.0.0.THEN.X.ELSE.0.0
RS/F8.0=ASAL*NOTA+ESAL*TCHRHIRES
FE/F8.0=RS*(FBR/100.0)
SE/F8.0=RS*(SAR/100.0)
END
TABLE
FILE PROGRAMS
PRINT RS AND FE AND SE AND SAVE
BY LV BY PID BY YOP
IF L IS &2
```

(1)

```
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/1.5.4.8.8.8
ORDER/30.34.10.57.58.59
SET/57.58.59
FILE PROGRAMS
END
```

```
DEFINE
FILE PROGRAMS
TOT-TCH-EXP/F8.0=BSAL+FRGF+SUME
END
TABLE
HEADING
&1
TEACHING EXPENSE
.
FILE PROGRAMS
PRINT ESAL AND ASAL AND BSAL AND FRGF AND SUME AND TOT-TCH-EXP
BY LV BY PID BY PN BY YOP
IF L IS &2
END
RETYPE
TABLE
HEADING
&1
TEACHING EXPENSE
.
FILE PROGRAMS
SUM BSAL AND FRGF AND SUME AND TOT-TCH-EXP
BY LV BY YOP BY PTYP
IF L IS &2
END
RETYPE
TABLE
HEADING
&1
TEACHING EXPENSE - SUBSIDIARY DATA
.
FILE PROGRAMS
PRINT ASII AND ESII AND FRW AND SAM
IF L IS &2
END
RETYPE
```

(19)

```
DEFINE
FILE PROGRAMS
CFY1/I4=CFY+1
CFY2/I4=CFY+2
CFY3/I4=CFY+3
CFY4/I4=CFY+4
CFY5/I4=CFY+5
END
TABLE
```

(20)

```
MIGM     DATA     P1    ID=GSS                04/13/73    11.17.30          PAGE 1


FILE PROGRAMS
PRINT CFY1 AND MSCPS AND TCPS AND CFY2 AND MSCPS AND TCPS AND
      CFY3 AND MSCPS AND TCPS AND CFY4 AND MSCPS AND TCPS AND
      CFY5 AND MSCPS AND TCPS AND SAVE
BY LV BY PID
IF YOP IS &3
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/1.5.4.7.7.C4.C7.C7.C4.C7.C7.C4.C7.C7.C4.C7.C7
ORDER/30.34.10.45.46.10.45.46.10.45.46.10.45.46.10.45.46
SET/45.46
FILE PROGRAMS
END
DEFINE
FILE PROGRAMS
MS/F7.2=MSCPS*(1.0+(MCRI/100.0))**(YOP-CFY)
TC/F7.2=TCPS*(1.0+(TRVR/100.0))**(YOP-CFY)
END
TABLE
FILE PROGRAMS
PRINT MS AND TC AND SAVE
BY YOP BY LV BY PID
IF YOP IS-NOT &3
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.1.5.7.7
ORDER/10.30.34.45.46
SET/45.46
FILE PROGRAMS
END
&& IF &4 LT 1 THEN GOTO LABEL2:
DEFINE
FILE PROGRAMS
EPRSTU/F8.0=(CSTU*PPW*CLT)/(NWO*NSP)
IPRSTU/I8=EPRSTU
END
TABLE
FILE PROGRAMS
WRITE IPRSTU AND SAVE BY LV BY CTY BY YOP
IF L IS &2
IF PTYP IS V
IF CTY IS-NOT V
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.67=1
FORM/1.1.4.4
ORDER/30.34.10.69
```

(21)

```
BIGM     DATA     P1     ID=GSS              04/13/73    11.17.39          PAGE

SET/69
FILE PROGRAMS
END
-------------------------------------------------------------------------------
&&LABEL2
-------------------------------------------------------------------------------
DEFINE
FILE PROGRAMS
MSE/F8.0=MSCPS*PSTU
TVE/F8.0=TCPS*PSTU
END
TABLE
FILE PROGRAMS
WRITE MSE AND TVE AND SAVE                                          (22)
BY YOP BY LV BY PID
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.1.5.8.8
ORDER/10.30.34.60.61
SET/60.61
FILE PROGRAMS
END
-------------------------------------------------------------------------------
TABLE
HEADING
&1
MATERIAL-SUPPLIES AND TRAVEL EXPENSES
.
FILE PROGRAMS
PRINT MSCPS AND TCPS AND MTSE AND TRVE
BY LV BY PID BY PN BY YOP
END
RETYPE                                                              (23)
TABLE
HEADING
&1
MATERIAL-SUPPLIES AND TRAVEL EXPENSES
.
FILE PROGRAMS
WRITE MTSE AND TRVE
BY YOP BY PTYP
END
RETYPE
-------------------------------------------------------------------------------
DEFINE
FILE PROGRAMS
TEQ/F8.0=CNE*REC
END
TABLE
FILE PROGRAMS                                                       (24)
WRITE TEQ AND HOLD BY YOP
IF L IS &2
IF PTYP IS V
END
DEFINE
FILE HOLD
```

```
CTEQ/FR.0=CTEQ+E02
END
TABLE
FILE HOLD
PRINT E01 AND CTEQ AND SAVE
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.R
ORDER/10.11
SET/11
FILE PROGRAMS
END
DEFINE
FILE PROGRAMS
MAINT-EX/FR.0=VVE*(EMCR/100.0)
END
TABLE
FILE PROGRAMS
WRITE FST*VVE AND FST*EMCR AND FST*MAINT-EX AND PSTU AND HOLD
BY YOP
IF L IS &2
IF PTYP IS V
END
DEFINE
FILE HOLD
MAIN-EX-PP/FR.2=E04/F05
END
TABLE
HEADING
&1
VOCATIONAL EQUIPMENT ON-HAND AND MAINTENANCE EXPENSE
.
FILE HOLD
PRINT E02 AND E03 AND E04 AND E05 AND MAIN-EX-PP BY E01
END
RETYPE
TABLE
FILE HOLD
PRINT MAIN-EX-PP AND SAVE
BY E01
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.X1.7
ORDER/10.15
SET/18
FILE PROGRAMS
END
DEFINE
FILE PROGRAMS
```

(25)

(26)

(27)

```
MAIN-EXP/FR.0=MAIPP*PSTU
END
TABLE
FILE PROGRAMS
WRITE MAIN-EXP AND SAVE BY YOP BY LV BY PID
IF L IS &2
IF PTYP IS V
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.1.5.R
ORDER/10.30.34.62
SET/62
FILE PROGRAMS
END
```

```
DEFINE
FILE PROGRAMS
TOT-TCHR/FR.0=BSAL+FRGE+SUBE
TOT-COST/FR.0=BSAL+FRGE+SUBE+MTSE+TRVE+MAIE+CNE+REC
END
TABLE
HEADING
&1
TOTAL PROGRAM COST NOT INCLUDING OVERHEAD
.
FILE PROGRAMS
PRINT TOT-TCHR AND MTSE AND TRVE AND MAIE AND CNE AND REC AND
      TOT-COST
BY PID BY PN BY YOP
IF L IS &2
END
RETY-E
```

(28)

```
DEFINE
FILE PROGRAMS
WSTU/F7.1=0WF*PSTU
END
TABLE
FILE PROGRAMS
WRITE WSTU AND SAVE
BY YOP BY OADC
IF OABST IS S
IF L IS &2
IF PTYP IS V
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.7.X-7.7
ORDER/10.21.26.20
SET/20.26
FILE PROGRAMS
END
```

(25)

```
BIGM     DATA     P1     ID=GSS              04/13/73    11.17.30          PAGE 1

DEFINE
FILE PROGRAMS
TOT-TCHR/F8.0=BSAL+FRGE+SUBE
TOT-COST/F8.0=BSAL+FRGE+SUBE+MTSE+TRVE+MAIE+CNE+REC
END
TABLE
HEADING
&1
TOTAL PROGRAM COST NOT INCLUDING OVERHEAD
.
FILE OUTPUT
COST-PER-STU/F8.2=C07/C08
RUN
FILE PROGRAMS
WRITE TOT-TCHR AND MTSE AND TRVE AND MAIE AND CNE AND REC AND
      TOT-COST AND FST*VSTU
BY YOP
IF L IS &2
END
RETYPE
```

(30)

```
TABLE
FILE PROGRAMS
PRINT TFTS AND SAVE BY YOP BY OAOC
IF OABST IS S
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.7
ORDER/10.21.25
SET/25
FILE PROGRAMS
END
```

(31)

```
TABLE
FILE PROGRAMS
PRINT TSQFT AND SAVE BY YOP BY OAOC
IF OABST IS Q
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.8
ORDER/10.21.25
SET/25
FILE PROGRAMS
END
```

(32)

```
TABLE
FILE PROGRAMS
PRINT TSTAF AND SAVE BY YOP BY OAOC
IF OABST IS T
IF L IS &2
```

(33)

```
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.7
ORDER/10.21.25
SET/25
FILE PROGRAMS
END
&& IF &6 LT 1 THEN GOTO LABEL3:
TABLEF
FILE PROGRAMS
PRINT YOP AND SAVE
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.21=&7.24=X
FORM/4
ORDER/10
SET/24
FILE PROGRAMS
END
&& IF &6 LT 2 THEN GOTO LABEL3:
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.21=&4.24=X
FORM/4
ORDER/10
SET/24
FILE PROGRAMS
END
&& IF &6 LT 3 THEN GOTO LABEL3:
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.21=&11.24=X
FORM/4
ORDER/10
SET/24
FILE PROGRAMS
END
&&LABEL3
TABLE
HEADING
&1
INPUT DATA - TOTAL LFA COSTS
.
FILE PROGRAMS
WRITE OAMTL AND COLUMN-TOTAL ACROSS YOP BY OADC BY OAP
IF L IS &2
END
RETYPE
```

(34)

(34)

(34)

(35)

```
MISM     DATA     P1     ID=GSS                04/13/73     11.17.30          PAGE 1

DEFINE
FILE PROGRAMS
D0200A/F8.0=BSAL + FRGE + SURE + MTSF + TRVF
D0700/F8.0=REC + MAIE
END
TABLE
FILE PROGRAMS
WRITE FST*OAMTL AND D0200A AND D0700 AND CNE AND HOLD
BY YOP BY OAOC
IF OAOC IS 0200A OR 0700 OR 1200
IF YOP IS &3
IF L IS &2
END
DEFINE
FILE HOLD
ROAMTL/F9.0=.IF.E02.EQ.'0200A'.THEN.E03-E04.ELSE.
     (.IF.E02.EQ.'0700'.THEN.E03-E05.ELSE.E03-E06)
END
TABLE
FILE HOLD
PRINT E01 AND E02 AND ROAMTL AND SAVE
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.X3.9
ORDER/10.21.28
SET/28
FILE PROGRAMS
END
-------------------------------------------------------------------------------
TABLE
HEADING
&1
TOTAL LEA COSTS LESS CURRENT YEAR DIRECT VOCATIONAL COSTS AND RELATED CO
.

URSES

FILE PROGRAMS
WRITE OAMTL AND COLUMN-TOTAL ACROSS YOP BY OAOC BY OA
IF L IS &2
END
RETYPE
-------------------------------------------------------------------------------
DEFINE
FILE PROGRAMS
TAPB/F8.2=OAMTL/OABSL
XAPB/F8.2=.IF.TAPB.GE.0.0.THEN.TAPB.ELSE.0.0
END
TABLE
FILE PROGRAMS
PRINT XAPB AND SAVE BY OAOC
IF YOP IS &3
IF OABST IS S OR Q OR T
IF L IS &2
```

(36)

(37)

(38)

```
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.10=&3
FORM/5.8
ORDER/21.27
SET/27
FILE PROGRAMS
END
```

```
DEFINE
FILE PROGRAMS
CFY1/I4=CFY+1
CFY2/I4=CFY+2
CFY3/I4=CFY+3
CFY4/I4=CFY+4
CFY5/I4=CFY+5
END
TABLE
FILE PROGRAMS
PRINT CFY1 AND OAAPB AND CFY2 AND OAAPB AND CFY3 AND OAAPB AND
      CFY4 AND OAAPB AND CFY5 AND OAAPB AND SAVE
BY OAOC
IF YOP IS &3
IF OABST IS S OR Q OR T
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/5.4.8.C4.C8.C4.C8.C4.C8.C4.C8
ORDER/21.10.27.10.27.10.27.10.27.10.27
SET/27
FILE PROGRAMS
END
```

(39)

```
DEFINE
FILE PROGRAMS
XAPB/F8.2=OAAPB*(1.0+(OAIR/100.0))**(YOP-CFY)
XAMT/F9.0=XAPB*OABSL
END
TABLE
FILE PROGRAMS
PRINT XAPB AND XAMT AND SAVE
BY YOP BY OAOC
IF YOP IS-NOT &3
IF OABST IS S OR Q OR T
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.8.9
ORDER/10.21.27.28
```

(40)

```
SET/27.2R
FILE PROGRAMS
END
TABLE
HEADING
&1
LEA COSTS LESS DIRECT VOC. COSTS (INCLUDING RELATED COURSES) NEW BASE
.
FILE PROGRAMS
WRITE &&PR ACROSS VOP MY &&OC MY DA. MY DA1= MY &&MST
IF &&RST IS S OR Q OR T
IF L IS &2
END
RETYPE
&& IF &A LT 1 THEN GOTO LAB=L&:
TABLEF
FILE PROGRAMS
PRINT VOP AND SAVE
END
UPDATE
READ/SAVE FT04F001
RECORD/45
INSERT/1=&2.3=&3.21=&7.24=&-
FORM/4
ORDER/10
SET/24
FILE PROGRAMS
END
&& IF &6 LT 2 THEN GOTO LAB=L&:
UPDATE
READ/SAVE FT04F001
RECORD/46
INSERT/1=&2.3=&3.21=&10.24=&10
FORM/4
ORDER/10
SET/24
FILE PROGRAMS
END
&& IF &6 LT 3 THEN GOTO LAB=L&:
UPDATE
READ/SAVE FT04F001
RECORD/46
INSERT/1=&2.3=&3.21=&11.24=&12
FORM/4
ORDER/10
SET/24
FILE PROGRAMS
END
&&LABEL&
TABLE
HEADING
&1
LEA COSTS LESS DIRECT VOCATIONAL COSTS AND RELATED COURSES
.
FILE PROGRAMS
```

(41) (42) (42) (42) (41)

```
MIDV      DATA      P1      ID=USS 1                    04/13/77     11.17.3P          PAGE
WRITE DAMIL AND COLUMN-TOTAL ACROSS YOP MY DAOC MY DA
IF L IS &2
END
RETYPE
-------------------------------------------------------------------------------
TABLE
FILE PROGRAMS
WRITE FST*VASTF AND NOTE AND HOLD
BY YOP BY DAOC
IF DABST IS T
IF L IS &2
IF PTYP IS V
END
DEFINE
FILE HOLD
TVS/F7.1=E03*E04
END
TABLE
FILE HOLD
PRINT E01 AND E02 AND E03 AND E04 AND TVS
   AND SAVE
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.X15.7
ORDER/20.21.26
SET/26
FILE PROGRAMS
END
-------------------------------------------------------------------------------
DEFINE
FILE SPACE
VSPC/I8=STAV*SFPS
END
TABLE
FILE SPACE
WRITE VSPC AND SAVE
BY YOP
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/48
INSERT/1=&2.3=&3
FORM/4.8
ORDER/10.19
SET/19
FILE PROGRAMS
END
TABLE
FILE PROGRAMS
PRINT VASP AND VTSP AND HOLD BY YOP
BY DAOC
IF DABST IS 0
IF L IS &2
```

```
END
DEFINE
FILE HOLD
TSV/IR=E03+E04
END
TABLE
FILE HOLD
PRINT E01 AND E02 AND E03 AND E04 AND TSV AND SAVE
END
UPDATE
READ/SAVE FT04F001
RECORD/9A
INSERT/1=&2.3=&3
FORM/4.5.X19.A
UNDER/10.21.24
SET/26
FILE PROGRAMS
END
```

```
DEFINE
FILE PROGRAMS
TOTSTUPROS/F4.0=IFTS*NEC*NSP
END
TABLE
FILE PROGRAMS
PRINT TOTSTUPROS AND SAVE AS YOP
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/9A
INSERT/1=&2.3=&3.21=020CA
FORM/4.9
UNDER/10.25
SET/25
FILE PROGRAMS
END
```

(46)

```
DEFINE
FILE PROGRAMS
TOTVOCPROS/F4.0=PSTU*[illegible]RNSP
END
TABLE
FILE PROGRAMS
WRITE TOTVOCPROS AND SAVE AS YOP
IF L IS &2
IF PTYP IS V
END
UPDATE
READ/SAVE FT04F001
RECORD/9A
INSERT/1=&2.3=&3.21=020CA
FORM/4.9
UNDER/10.25
SET/26
FILE PROGRAMS
END
```

(47)

```
BIGM     DATA     P1     ID=GSS          .     04/13/73     11.17.30          PAG
```

```
DEFINE
FILE PROGRAMS
SPRDS/F8.1=CSTU*PPW*CLT
END
TABLE
FILE PROGRAMS
WRITE FST*OABSL AND FST*OABSV AND SPRDS AND HOLD BY YOP
IF OAOC IS 0200A
IF L IS &2
IF PTYP IS V
END
DEFINE
FILE HOLD
REMTOTPRDS/F9.0=E02-E04
REMVOCPRDS/F9.0=E03-E04
END
TABLE
FILE HOLD
PRINT E01 AND REMTOTPRDS AND REMVOCPRDS AND SAVE
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3.21=0200A
FORM/4.9.9
ORDER/10.25.26
SET/25.26
FILE PROGRAMS
END
```

(48)

```
&& IF &S GT 0 THEN GOTO LABEL5:
DEFINE
FILE PROGRAMS
XAMT/F9.0=(OABSV/OABSL)*OAMTL
END
```

(49)

```
&& GOTO LABEL6
&&LABEL5
DEFINE
FILE PROGRAMS
XAMT/F9.0=OAMTL
END
```

(50)

```
&&LABEL6
TABLE
FILE PROGRAMS
PRINT XAMT AND SAVE BY YOP BY OAOC
IF OABST IS S OR Q OR T
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.9
ORDER/10.21.29
SET/29
FILE PROGRAMS
```

(51)

```
END
DEFINE
FILE PROGRAMS
D0200A/FR.0=BSAL + FRGE + SURE + MTSE + TRVF
D0700/FR.0=REC + MAIF
END
TABLE
FILE PROGRAMS
WRITE FST*OAMTL AND D0200A AND D0700 AND CNE AND HOLD
BY YOP BY OAOC
IF OAOC IS 0200A OR 0700 OR 1200
IF L IS &2
END
DEFINE
FILE HOLD
ROAMTL/F9.0=.IF.E02.EQ.'0200A'.THEN.E03*E04.ELSE.
     (.IF.E02.EQ.'0700'.THEN.E03*E05.ELSE.E03*E06)
END
TABLE
FILE HOLD
PRINT E01 AND E02 AND ROAMTL AND SAVE
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.5.X3.9
ORDER/10.21.28
SET/28
FILE PROGRAMS
END
TABLE
HEADING
&1
TOTAL LEA COSTS WITH DIRECT VOCATIONAL COSTS AND RELATED COURSES
.
FILE PROGRAMS
WRITE OAMTL AND COLUMN-TOTAL ACROSS YOP BY OAOC BY OAM
IF L IS &2
END
RETYPE
TABLE
HEADING
&1
LEA COSTS ALLOCATED TO VOCATIONAL PROGRAMS
.
FILE PROGRAMS
WRITE OAMTV AND COLUMN-TOTAL ACROSS YOP BY OAOC BY OAM
IF L IS &2
END
RETYPE
DEFINE
FILE PROGRAMS
OMPS/F7.2=OAMTV/VSTU
END
```

(52)

(51)

```
TABLE
FILE PROGRAMS
WRITE OMPS AND SAVE BY YOP
IF OABST IS S OR Q OR T
IF L IS &2
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.7
ORDER/10.18
SET/18
FILE PROGRAMS
END
```

(54)

```
DEFINE
FILE PROGRAMS
OVHC/FR.0=MAIPP*PSTU
END
TABLE
FILE PROGRAMS
WRITE OVHC AND SAVE BY YOP BY LV BY PID
IF L IS &2
IF PTYP IS V
END
UPDATE
READ/SAVE FT04F001
RECORD/96
INSERT/1=&2.3=&3
FORM/4.1.5.8
ORDER/10.30.34.63
SET/63
FILE PROGRAMS
END
```

(55)

```
DEFINE
FILE PROGRAMS
TOT-TCHR/FR.0=BSAL+FRGE+SUBE
MTS-TRV/FR.0=MTSE+TRVE
EQPT-COST/FR.0=CNE+REC
TOT-COST/FR.0=BSAL+FRGE+SUBE+MTSE+TRVE+MAIE+CNE+REC+OVHE
END
TABLE
HEADING
&1
TOTAL VOCATIONAL PROGRAM COST WITH OVERHEAD
.
FILE PROGRAMS
PRINT TOT-TCHR AND MTS-TRV AND MAIE AND EQPT-COST AND OVRE AND
      TOT-COST
BY LV BY PID BY PN BY YOP
IF L IS &2
END
RETYPE
TABLE
HEADING
```

(56)

```
BIGM     DATA     P1     ID=GSS                  04/13/73     11.17.30          PAGE

&1
TOTAL VOCATIONAL PROGRAM COST WITH OVERHEAD
.
FILE OUTPUT
COST-PER-STU/F8.2=C06/C07
RUN
FILE PROGRAMS
WRITE TOT-TCHR AND MTS-TRV AND MATE AND EQPT-COST AND OVPE AND
      TOT-COST AND FST*VSTU
BY YOP
IF L IS &2
END
RETYPE
-------------------------------------------------------------------------------
TABLE
HEADING
&1
TOTAL STUDENTS AND TOTAL VOCATIONAL STUDENTS
.
FILE OUTPUT
PCT-VOC/F5.1=(C02/C01)*100.0
RUN
FILE PROGRAMS
PRINT TFTS AND VSTU BY YOP
IF L IS &2
END
RETYPE
DEFINE
FILE PROGRAMS
TOT-SPACE/F8.0=OABSL
VOC-SPACE/F8.0=OABSV
END
TABLE
HEADING
&1
TOTAL SPACE AND VOCATIONAL SPACE IN SQUARE FEET
.
FILE OUTPUT
PCT-VOC/F5.1=(C02/C01)*100.0
RUN
FILE PROGRAMS
PRINT TOT-SPACE AND VOC-SPACE BY YOP
IF OAOC IS 0700
IF L IS &2
END
RETYPE
DEFINE
FILE PROGRAMS
TOT-STAFF/F8.1=OABSL
VOC-STAFF/F8.1=OABSV
END
TABLE
HEADING
&1
TOTAL STAFF AND VOCATIONAL STAFF
.
```

(57)

```
BIGM     DATA     P1     ID=GSS                04/13/73     11.17.30          PAGE

FILE OUTPUT
PCT-VOC/F5.1=(C02/C01)*100.0
RUN
FILE PROGRAMS
PRINT TOT-STAFF AND VOC-STAFF BY YOP
IF 0A0C IS 0800A
IF L IS &2
END
RETYPE
DEFINE
FILE PROGRAMS
TOT-STU-PRDS/F10.0=0ARSL
REM-STU-PRDS/F10.0=0ARSV
END
TABLE
HEADING
&1
TOTAL STUDENT PERIODS AND VOC. STUDENT PERIODS IN NON-VOC.. ACAD.. AND R
.

ELATED COURSES

FILE OUTPUT
PCT-REM/F5.1=(C02/C01)*100.0
RUN
FILE PROGRAMS
PRINT TOT-STU-PRDS AND REM-STU-PRDS BY YOP
IF 0A0C IS 0200A
IF L IS &2
END
RETYPE
FINISH
```

10.2 Edit BIGM DATA to insert the appropriate parameter values and option flags.

Parameter	Description
&1	LEA and date title used as first line of the heading for each report
&2	6 character LEA code
&3	Current year (e.g., 1972)
&4	1 if there are related/academic programs 0 if there are no related/academic programs
&5	1 if all the students are vocational students 0 if all the students are not vocational students
&6	Number of overhead accounts that are not to be projected over time but are to be allocated to vocational programs (0, 1, 2, or 3)
&7	First overhead account code
&8	First overhead account base (S, T, or Q)
&9	Second overhead account code
&10	Second overhead account base (S, T, or Q)

&11	Third overhead account code
&12	Third overhead account base (S, T, or Q)

Insert the title on the first line of BIGM DATA between the quotes (because of blanks in the title) and make sure the line ends with a comma so the Run Executive knows the EXEC card is continued onto the second line. Change the second line for the appropriate LEA code, current year, related/academic flag (1 or 0), all vocational student flag (1 or 0) and overhead account data.

If no overhead account, which is to be allocated to vocational programs, is input over time, then put a 0 after the two flags. If there are allocatable overhead accounts which are input over time, then put the number of these accounts after the flags followed by account, base, account, base, etc. as many as necessary (up to 3 accounts). If more than 3 accounts are needed, then BIGM DATA has to be edited for additional procedures (34) and (42).

The third line of BIGM DATA (&&ARGS) is used to assign default overhead account codes and bases. These default values are not used but are necessary so the Run Executive will not fail due to missing parameters if less than three allocatable overhead accounts are specified. If any of the title, LEA, current year, or flags are missing, the Run Executive will not execute.

10.3 The model uses the CSS file SAVE FT04001 on U-disk as a temporary file for spin-offs. If a SAVE FT04F001 file exists on P-disk, the model will run incorrectly because the model will write onto U-disk but read from P-disk. Before putting the batbh job into the queue, type "erase save ft04f001" in CSS mode.

10.4 The following CSS file called BATCH EXEC should be created. BATCH EXEC contains the CSS commands which will be executed when the batch job emerges from the queue for execution. The ROUTE command in this example assumes Philadelphia.

To place the job in the batch queue, type in CSS mode "cssbatch 2 time 0 account # core 512". Priority 2 is not guaranteed overnight turnaround but in the past outputs have always been available the next day. Maximum time estimates used have been between 300 to 1500 seconds depending upon the size of the LEA in number of programs and courses.

11.0 Re-running the Model.

Review of the initial run of the model usually results in changes to be made in the data and the request for additional runs. The following steps should be executed before any additional run.

11.1 Remove the spin-off from the PROGRAMS file to the SPACE file. Execute the following RAMIS procedure:

```
SCAN
FILE SPACE
SHOW 1, 3, 4, 5, 13
LOCATE 13 = level code,$1
DELETE *
END
```

11.2 If the model was run using related/academic programs, the spin-off from the PROGRAMS file to itself (to put related/academic periods of vocational students under the related/academic programs) must be removed. Execute the following RAMIS procedure:

```
SCAN
FILE PROGRAMS
SHOW 1, 3, 10, 30, 34, 67
LOCATE 36=A, 67=1,$1        } if there are
DELETE *                    } academic programs
```

```
TOP
LOCATE 36=R, 67=1,$1   }  if there are
DELETE *               }  related programs
END
```

11.3 Edit INP2 DATA to change the procedures for the input of overhead data. Change procedure (6) so that SET/21, 22, 23, 24, 28 becomes SET/22, 23, 24, 28. Change procedure (7) so that the INCLUDE becomes an UPDATE and SET/21, 22, 23, 24, 28 becomes SET/22, 23, 24, 28. The following listing shows the edited INP2 DATA.

11.4 Execute INP2 DATA by typing "ramisin inp2 data" in the CSS mode.

11.5 Run the model as described in Section 10.0.

p inp2 data

```
UPDATE
FILE MASTER
1=PROGRAMS,5=VVE,11=SET=0,$
END
```

(1)

```
UPDATE
FILE PROGRAMS
1=233150,2=SET=MIDDLESEX COUNTY VOCATIONAL SCHOOLS,3=1972,
4=SET=10,5=SET=10,6=SET=15,7=SET=4.21,8=SET=10,9=SET=8,$
END
```

(2)

```
UPDATE
INSERT/1=233150,3=1972
FORM/4,7,6,5,2,5,4,4
ORDER/10,11,12,13,14,15,16,17
SET/11,12,13,14,15,16,17
FILE PROGRAMS
19723326370257163120440519 .5191377.5
19733326370257163120440519 5.0210080.0
19743326370257163120440519 8.0220084.0
19753326370257163120440520 0.0240084.0
19763326370257163120440620 0.0260085.0
19773326370257163120440720 0.0280085.0
END
```

(3)

```
UPDATE
INSERT/1=233150,3=1972,30=S
FORM/4,2,2,3
ORDER/10,31,32,33
SET/31,32,33
FILE PROGRAMS
197238401.0
197338401.0
197438401.0
197538401.0
197638401.0
197738401.0
END
```

(4)

```
UPDATE
FILE MASTER
1=PROGRAMS,5=VVE,11=SET=1,$
END
```

(5)

```
UPDATE
READ/OVH1 DATA
INSERT/1=233150,3=1972
FORM/4,5,20,4,1,7
ORDER/10,21,22,23,24,28
SET/22,23,24,28
FILE PROGRAMS
END
```

(6)

```
UPDATE
READ/OVH2 DATA
INSERT/1=233150,3=1972
FORM/4,5,20,4,1,7
ORDER/10,21,22,23,24,28
SET/22,23,24,28
FILE PROGRAMS
END
FINISH
```

(7)

14.36.23

Appendix

Annotated File Descriptions for

FILE PROGRAMS

FILE SPACE

FILE TMPCSE

NUMBER OF RECORDS IN TABLE= 78 LINES= 78

PAGE 1

DESCRIPTION FOR FILE PROGRAMS

LIST	FIELD NAME	SYNONYM	TYPE	LEVEL	SEGMENT	DESCRIPTION
1	LEA-NUMBER	L	A6	1	4	LEA number in the form of 2-digit county code followed by 4-digit LEA code within county, e.g., 354180.
2	LEA-NAME	LN	A60	1	4	LEA name.
3	CUR-FISC-YR	CFY	I4	2	2	Current year, e.g., planning cycle occurring in 1971-1972 school year would have the current year as 1972 and first year being planned is 1972-1973 (1973).
4	V-TCHR-ATTR	VTAR	P4.1	2	2	Vocational teacher attrition rate in percent.
5	N-TCHR-ATTR	NTAR	P4.1	2	2	Non-vocational teacher attrition rate in percent.
6	FRNG-RATE	FBR	P4.1	2	2	Fringe benefit cost as percent of base salary cost.
7	SUBST-ALLOW	SAR	P4.1	2	2	Substitution cost as percent of base salary cost.
8	AVER-SAL-INC	ASII	P4.1	2	2	Yearly percent change in average salary per teacher.
9	ENTR-SAL-INC	ESII	P4.1	2	2	Yearly percent change in entering salary per teacher.
10	PLAN-YEAR	YOP	I4	3	6	Year of plan, either current year or future year, e.g., 1972, 1973, etc.
11	VOC-EQUIP	VVE	I8	3	6	Total dollar value of vocational equipment, not including equipment under programs.
12	TOT-SQFT	TSQFT	I8	3	6	Total square feet in LEA for all purposes.
13	VOC-ADMIN-FT	VASP	I8	3	6	Total square feet in LEA for vocational administrative purposes.
14	PCT-EQP-MNT	EMCR	P4.1	3	6	Percent maintenance cost to total dollar value of equipment.
15	TOTAL-STAFF	TSTAF	P7.1	3	6	Total staff in LEA.
16	FTE-STUD	TFTS	P7.1	3	6	Total full-time equivalent students in LEA.
17	VOC-ADM-STAF	VASTF	P7.1	3	6	Total staff in LEA administering vocational programs.
18	MOOV-PER-STU	MAIPP	P7.2	3	6	Maintenance or overhead cost per student.
19	VOC-INS-FT	VTSF	I8	3	6	Total vocational instructional square feet.
20	VOC-STUD	VSTU	P7.1	3	6	Total vocational students.
21	OVER-ACT-NO	OAOC	A5	4	-19	Overhead account code.
22	OVER-ACT-NAM	OAN	A20	4	-19	Overhead account name.
23	OVER-INFL-RT	OAIR	P4.1	4	-19	Overhead account yearly percent change.
24	OVER-BASE-TY	OABST	A1	4	-19	Type of base used to allocate overhead cost to vocational program (S=students, T=staff, Q=square feet, x=not allocate).
25	OVER-BASE-L	OABSL	F9.1	4	-19	Total LEA base for overhead account.
26	OVER-BASE-V	OABSV	F9.1	4	-19	Vocational base for overhead account.

DESCRIPTION FOR FILE PROGRAMS

LIST	FIELD NAME	SYNONYM	TYPE	LEVEL	SEGMENT	DESCRIPTION
27	AMT-PER-BASE	OAAPB	F8.2	4	-19	Overhead account cost per LEA base.
28	OVER-AMT-T	OAMTL	F9.0	4	-19	Overhead account cost for LEA.
29	OVER-AMT-V	OAMV	F9.0	4	-19	Overhead account cost for vocational programs
30	LEVEL	LV	A1	5	2	Level code (S=secondary, P=post-secondary, A=adult)
31	WEEKS-OPEN	NWO	F4.1	5	2	Number of weeks per year schools open.
32	PDS-PER-WEEK	NSP	F4.1	5	2	Number of periods per week.
33	OVER-WGT-FAC	OWF	F5.2	5	2	Overhead weighting factor, not used, should be set to 1.0.
34	PROG-IDEN	PID	A5	6	10	Program code.
35	PROG-STATUS	PSC	A4	6	10	Program status, not used, originally intended to indicate whether program in the base case on alternative case, or not.
36	PROG-TYPE	PTYP	A1	6	10	Program type (V=vocational, R=related, A=academic) (Mathematics program, Science Program, etc. should have program type either R or A).
37	PROG-CAT	PCAT	A1	6	10	Program category, not used.
38	MTLS-RATE	MCRI	F4.1	6	10	Yearly percent change for materials/supplies cost per student.
39	TRAVEL-RATE	TRVR	F4.1	6	10	Yearly percent change for travel cost per student.
40	PROG-NAME	PN	A25	6	10	Program name.
41	TCHR-REQ	NOTR	F5.1	6	10	Number of full-time equivalent teachers required.
42	TCHR-AVL	NOTA	F5.1	6	10	Number of full-time equivalent teachers available.
43	AVER-SAL	ASAL	I7	6	10	Average salary per teacher.
44	ENT-SAL	ESAL	I6	6	10	Entering salary per teacher.
45	MTLS-COST-ST	MSCPS	F7.2	6	10	Materials/Supplies cost per student.
46	TRAV-COST-ST	TCPS	F7.2	6	10	Travel cost per student.
47	COMP-RATE	CMPR	F5.1	6	10	Percent students in last year of program completing program.
48	PLACE-RATE	PLER	F5.1	6	10	Percent students completing program able to be placed.
49	COUN-MAN-POR	CPMN	I6	6	10	Manpower requirement forecast for county of LEA.
50	LEA-MAN-PCT	PLCP	F5.1	6	10	Percent of county requirement allocated to LEA.
51	AVER-CL-SIZE	ACSD	F5.1	6	10	Average class-size.
52	TCHR-PDS	TAVP	F4.1	6	10	Number of teacher periods per week.

DESCRIPTION FOR FILE PROGRAMS

LIST	FIELD NAME	SYNONYM	TYPE	LEVEL	SEGMENT	DESCRIPTION
53	TCHR-WKS	TAVW	P4.1	6	10	Number of teacher weeks per year.
54	NEW-EQUIP	CNE	P8.0	6	10	Dollar value of new equipment.
55	REPL-EQUIP	REC	P8.0	6	10	Dollar value of replacement equipment.
56	SOC-DEMAND	SOCD	I6	6	10	Social demand - number of students wanting to enter first year of program.
57	BASE-SALARY	BSAL	P8.0	6	10	Base salary cost.
58	FRINGE-BENE	FRGE	P8.0	6	10	Fringe benefit cost.
59	SUBST-EXP	SUBE	P8.0	6	10	Substitution cost.
60	MAT-AND-SUP	MTSE	P8.0	6	10	Materials/Supplies cost.
61	TRAVEL	TRVE	P8.0	6	10	Travel cost.
62	MAINTENCE	MAIE	P8.0	6	10	Maintenance cost.
63	OVERHEAD	OVRE	P8.0	6	10	Overhead cost.
64	REVENUE-CODE	RVSC	A5	7	-2	Revenue source code - not used.
65	REVENUE-NAME	RVSN	A20	7	-2	Revenue source name - not used.
66	REV-PCT	PTPC	P5.1	7	-2	Percent of program cost to be applied against revenue source - not used.
67	PROG-YR	PYR	I1	8	3	Year of program (e.g., first year of program, second, etc.)
68	FINAL-YEAR	FYON	A1	8	3	Flag whether or not this year of program is the final year of the program.
69	STUD-IN-PROG	PSTU	I8	8	3	Number of full-time equivalent students in the program in the year of the program.
70	COURSE	CC	A6	9	3	Course code.
71	COURSE-TYPE	CTY	A1	9	3	Course type (V=Vocational, M=Mathematics, S=Science, etc.)
72	COURSE-NAME	CN	A20	9	3	Course name..
73	OE-CODE	OE	I8	9	3	OE code.
74	SPACE-TYPE	SPT	A3	9	3	Space type code (if no space type appropriate, use 000).
75	CRS-PDS-WK	PPW	P4.1	9	3	Course periods per week.
76	CRS-LENGTH	CLT	P4.1	9	3	Course length in weeks.
77	GRADE	G	I2	10	2	Grade code.
78	ENROLLMENT	CSTU	I4	10	2	Course enrollment.

NUMBER OF RECORDS IN TABLE= 21 LINES= 21

PAGE 1

DESCRIPTION FOR FILE SPACE

LIST	FIELD NAME	SYNONYM	TYPE	LEVEL	SEGMENT	DESCRIPTION
1	LEA-NUMBER	L	A6	1	4	LEA number.
2	LEA-NAME	LN	A60	1	4	LEA name.
3	CUR-FISC-YR	CFY	I4	2	2	Current year.
4	PLAN-YEAR	YOP	I4	3	6	Year of plan.
5	SPACE-TYPE	SPT	A3	4	10	Space type code.
6	SPACE-USE	SPU	A1	4	10	Space use code (V=vocational, R=related, A=academic).
7	FAC-AVAIL	FAV	F5.1	4	10	Number of facilities available.
8	STAT-AVAIL	STAV	F6.1	4	10	Number of stations available across all facilities.
9	SQ-FT-PER-ST	SFPS	F5.1	4	10	Square feet per station.
10	PDS-AVL-WK	PPWA	F4.1	4	10	Periods per week a facility is available.
11	STAT-UTILIZ	STUR	F5.1	4	10	Percent of a week a facility could be scheduled to be utilized because of scheduling impossibilities.
12	SPACE-NAME	SPN	A25	4	10	Space type name.
13	LEVEL	LV	A1	5	4	Level code.
14	WEEKS-OPEN	NWO	F4.1	5	4	Number of weeks per year LEA open.
15	PROG-IDEN	PID	A5	6	1	Program code.
16	PROG-STATUS	PSC	A4	6	1	Program status, see PROGRAMS file, not used.
17	PROG-NAME	PN	A25	6	1	Program name.
18	AVER-CL-SIZE	ACSD	F4.1	6	1	Average class-size.
19	PROG-YR	PYR	I1	7	3	Year of program.
20	COURSE-CODE	CC	A6	8	3	Course code.
21	TOT-STUD-PDS	SPRDS	F9.1	8	3	Total yearly student periods for course.

NUMBER OF RECORDS IN TABLE= 11 LINES= 11

PAGE 1

DESCRIPTION FOR FILE TMPCSE

LIST	FIELD NAME	SYNONYM	TYPE	LEVEL	SEGMENT	DESCRIPTION
1	CC	CC	A6	1	100	Course code.
2	YOP	YOP	I4	2	6	Year of plan.
3	CTY	CTY	A1	2	6	Course type.
4	CN	CN	A20	2	6	Course name.
5	OE	OE	I8	2	6	OE code.
6	SPT	SPT	A3	2	6	Space type (if no space type, use 000).
7	PPW	PPW	P4.1	2	6	Periods per week.
8	CLT	CLT	P4.1	2	6	Course length in weeks.
9	LV	LV	A1	3	1	Level.
10	PID	PID	A5	4	1	Program code.
11	PYR	PYR	I1	5	3	Year of program.